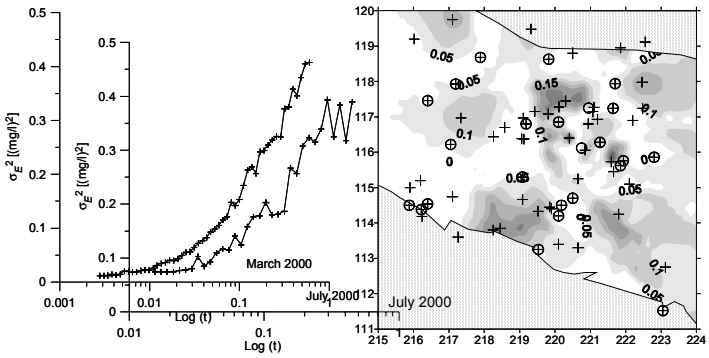
UNIVERSIDADE TÉCNICA DE LISBOA

INSTITUTO SUPERIOR TÉCNICO



MONITORING NETWORKS OPTIMIZATION WITH SIMULATED ANNEALING

Luís Miguel de Amorim Ferreira Fernandes Nunes
(Mestre)

Dissertação para obtenção do Grau de Doutor em Ciências de Engenharia

Orientador: Doutor Luís Filipe Tavares Ribeiro
Co-orientador: Doutora Maria da Conceição Morais de Oliveira Cunha

Presidente: Reitor da Universidade Técnica de Lisboa
Vogais: Doutor Henrique José Figueiredo Garcia Pereira
Doutor Pierre Yves Hubert
Doutor Fernando de Oliveira Durão
Doutor António Jorge Gonçalves de Sousa
Doutora Maria da Conceição Morais de Oliveira Cunha
Doutor Luís Filipe Tavares Ribeiro

Outubro de 2003

ABSTRACT

In this work some methods to optimize environmental monitoring networks are proposed. These methods share simulated annealing as the approximation algorithm.

Only monitoring networks reduction is treated here. Monitoring network optimization is a very actual problem given the large number of existing networks in many countries operating large numbers of stations, some of which may be redundant, with very high exploitation costs.

Difficulties appear when exploitation costs pushes the dimension of a network towards a minimum, and the statistical reliability pushes in the opposite direction. Finding the optimal dimension may be a very difficult optimization problem due to the large number of combinations, even for small network dimensions. Further complications appear when the available data is too incomplete or come from different homogeneous areas. Some practical answers to these problems were sought in this work.

Results showed that optimizing a monitoring network dimension and location of stations, without compromising the quality of the collected data, could attain large reductions in exploitation costs. Simulated annealing showed to be a very flexible and efficient algorithm.

Key-words: Monitoring network, optimization, combinatorial problems, simulated annealing.

RESUMO

Neste trabalho são propostos alguns métodos para optimização de redes de monitorização ambiental, os quais têm em comum o recozimento simulado como algoritmo de aproximação.

Tratam-se apenas problemas relacionados com a redução da dimensão de redes já existentes. Considera-se este assunto muito actual, dada a necessidade de racionalizar os custos das redes Europeias já em operação, algumas das quais com milhares de pontos, muitos dos quais redundantes.

Duas "forças" distintas operam durante a optimização – a redução de custos e a representatividade estatística. Enquanto a primeira é feita, em grande parte, por minimização da dimensão da rede, a segunda está usualmente ligada à maximização da dimensão. Encontrar o ponto de equilíbrio é um problema de difícil resolução dado o grande número de combinações possíveis entre as estações de monitorização, mesmo para redes de pequena dimensão. Maiores dificuldades aparecem ainda quando o conjunto de dados disponíveis é muito incompleto ou é aconselhável dividir a área em estudo em regiões homogéneas. Neste trabalho são apresentadas respostas práticas para estes problemas.

Os resultados mostraram que é possível reduzir a dimensão de redes existentes sem comprometer a representatividade estatística da informação recolhida, o que se reflecte em menores custos de exploração. O algoritmo de recozimento simulado mostrou ser muito flexível e eficiente na resolução destes problemas.

Palavras-chave: redes de monitorização, optimização, problemas combinatórios, recozimento simulado.

ACKNOWLEDGEMENTS

The author would like to acknowledge:

Prof. Maria da Conceição Cunha from the University of Coimbra and Prof. Luís Ribeiro from the Technical University of Lisbon for their support and timing corrections of the manuscript;

Annonymous reviewers of the *Environmental Monitoring and Assessment*, *Journal of Water Resources Planning and Management*, *Water Resources Research*, and *Advances in Water Research* whom with their corrections and proposals helped improving the texts of chapters three to six;

The co-authors of the articles, presented in chapters three to six for presenting me their problems, sharing their data, and helping in describing their case-studies;

My family for their support;

My wife for her patience and love.

TABLE OF CONTENTS

ABSTRACT ..... **V**
RESUMO ..... **VII**
ACKNOWLEDGEMENTS ..... **IX**
FIGURES ..... **XV**
TABLES ..... **XVII**
NOTATION ..... **XIX**
1 INTRODUCTION **1**
1.1 OBJECTIVES 1
1.2 JUSTIFICATION 2
1.3 WORK ORGANIZATION 5
2 BIBLIOGRAPHIC REVIEW **9**
2.1 MONITORING NETWORK OPTIMIZATION 9
2.2 STOCHASTIC OPTIMIZATION 16
2.2.1 Stochastic Methods 18
2.2.1.1 Two-phase methods *18*
2.2.1.2 Random search methods *23*
2.2.2 Simulated annealing 26
2.2.2.1 Finite-time approximation *28*
3 GROUNDWATER MONITORING NETWORKS OPTIMIZATION WITH REDUNDANCY REDUCTION **35**
3.1 SYLLABUS 35
3.2 INTRODUCTION 37
3.3 SPACE AND TIME MODELS FOR NETWORK OPTIMIZATION 41
3.3.1 Estimation variance 41
3.3.2 Redundancy 44
3.3.3 Resulting optimization model 45
3.4 SOLVING THE OPTIMIZATION PROBLEM 46
3.4.1 Iterative improvement 46
3.4.2 Analysis of the convergence 49
3.5 APPLICATION EXAMPLE 51
3.6 RESULTS AND DISCUSSION 53
3.6.1 Network optimization only with spatial information 55

3.6.2 Network optimization with only temporal information 58
3.6.3 Space-time network optimization 60
3.7 CONCLUSIONS 63
3.8 ANNEX TO SECTION 64
4 OPTIMAL SPACE-TIME COVERAGE AND EXPLOITATION COSTS IN GROUNDWATER MONITORING NETWORKS 67
4.1 SYLLABUS 67
4.2 INTRODUCTION 69
4.3 OBJECTIVE FUNCTION 73
4.3.1 Spatial accuracy 74
4.3.2 Temporal redundancy 78
4.3.3 Exploitation costs 79
4.3.4 Optimization model 79
4.4 SOLVING THE MODEL 80
4.5 CASE STUDY 85
4.6 RESULTS 86
4.6.1 Cost-benefit analysis 86
4.6.2 Selected monitoring network 90
4.7 CONCLUSIONS 92
5 GROUNDWATER NITRATE MONITORING NETWORK OPTIMIZATION WITH MISSING DATA 95
5.1 SYLLABUS 95
5.2 INTRODUCTION 97
5.3 CASE-STUDY 100
5.4 METHODS 102
5.4.1 Reducing the dimension of an existing groundwater monitoring network 102
5.4.2 Estimation error variance 104
5.4.3 Optimizing a monitoring network 106
5.5 APPLICATION TO THE CASE-STUDY 110
5.5.1 Selecting the optimal monitoring network 111
5.5.2 Validation of selected monitoring networks 122
5.6 CONCLUSIONS 126
6 OPTIMAL ESTUARINE SEDIMENT MONITORING NETWORK DESIGN WITH SIMULATED ANNEALING 129
6.1 SYLLABUS 129
6.2 INTRODUCTION 131
6.3 THEORY 132

6.3.1 Estimation of probability distribution functions 132
6.3.2 Estimation error and estimation error variance 135
6.3.3 Optimization function model 136
6.3.4 Solving the optimization function model 137
6.4 CASE STUDY 140
6.4.1 Study area and source data 140
6.4.2 Data processing 140
6.5 OPTIMIZATION RESULTS AND DISCUSSION 143
6.5.1 Feasible space 143
6.5.2 Optimal conditioning 146
6.5.3 Optimal monitoring network 152
6.6 CONCLUSIONS 152
7 WORK CONCLUSIONS 155
REFERENCES 161
APPENDIX A: FORTRAN 90 OPTIMIZATION CODES 185
CODE FOR MONITORING NETWORK OPTIMIZATION 185

FIGURES

Figure 1.1. Monitoring cycle 3

Figure 3.1. Simulated annealing algorithm. Random is a random real number taken from a uniform distribution in (0,1). 48

Figure 3.2. Minimizing only σ_E^2 : a) σ_E^2 (un^2); b) OF; c) H/H_{max} 56

Figure 3.3. Minimizing only σ_E^2 : optimal solution 57

Figure 3.4. Maximizing only S: a) S (un); b) OF; c) H/H_{max}. 59

Figure 3.5. Maximizing only S: optimal solution 60

Figure 3.6. Minimizing σ_E^2 and maximizing S: a) σ_E^2 (un^2); b) S (un); c) OF; d) H/H_{max} 62

Figure 3.7. Minimizing σ_E^2 and maximizing S: optimal solution 62

Figure 3.8. Experimental omnidirectional variogram and adjusted spherical model 65

Figure 4.1. Objective function computing module 83

Figure 4.2. Location of the monitoring stations and routes connecting them 86

Figure 4.3. Cost-benefit analysis for increasing number of stations: a) gain in spatial accuracy and temporal representativeness; b) Time necessary to sample 88

Figure 4.4. Relative contribution of UC to the total sampling time. 89

Figure 4.5. Stations excluded in the new designs 90

Figure 4.6. Optimization results for a monitoring network of 23 stations. a): Objective function; b) data quality variables; c) sampling times 91

Figure 4.7. Resulting monitoring network with 23 stations 91

Figure 5.1. Geographical location of Beja's Gabbro aquifer (350 km^2) and study area (50 km^2). 102

Figure 5.2. Proposed method 103

Figure 5.3. Projection of measuring times in the first two PCA factors of variogram parameters. Selected measuring times are identified 114

Figure 5.4. Objective function convergence curves ($x10^3$). Temperature in log scale. 116

Figure 5.5. Nitrate concentrations (mg/l) 118

Figure 5.6. Optimal spatial distribution of stations in the eight measuring times 120

Figure 5.7. Stations included in the GMN. With frequency of at least a) three stations; b) four stations. 122

Figure 5.8. Field estimated with raw data for a) Nov. 1999; Mean relative absolute errors with 24 stations for b) Nov. 1999; c) July 2000; idem with 16 stations for d) Nov 1999; e) July 2000 125

Figure 6.1. Flowchart for the definition of the sediment homogenous areas 142

Figure 6.2. Effect of conditioning and cardinality of S': a) Dimension of the solution space (W) and time until optimal solution (T); b) Number of iterations 145

Figure 6.3. Relative error when estimating $\Psi_{S'}(x_\alpha z_c)$*: a) no conditioning; b)* $\delta = 0.5$*; c)* $\delta = 0.3$ 147

Figure 6.4. Mean square estimation error, s^2*, without conditioning and with conditioning (*$\delta = 0.3$*;* $\delta = 0.5$*)* 148

Figure 6.5. Monitoring networks for different ω_{Max} *values: a) no conditioning (nc)* 149

TABLES

Table 3.1. Time series equations 52

Table 3.2. Solutions: Minimizing only σ_E^2, maximizing only S, and minimizing σ_E^2 and maximizing S. OF: objective function 58

Table 3.3 Data statistics and spherical variogram parameters 66

Table 4.1. Time series equations 85

Table 4.2. Optimization results for the eight ω values. The original GMN results are also shown 89

Table 5.1. Statistical parameters for the 19 measuring times (mg/l) 111

Table 5.2. Parameters of spherical space variograms models for the 19 measuring times 113

Table 5.3. Optimization results 116

Table 5.4. Mean relative absolute error (%) 123

Table 6.1. Physic-chemical sediment parameters of each homogeneous areas 141

Table 6.2. Indicators frequencies, $\phi(A,z_c)$ and $\Psi(A,z_c)$ 142

Table 6.3. Results for different ω_{Max} and δ values: mean square estimation error (s^2), number of iterations (Iter.) and time necessary to reach an optimal solution (T) in seconds 145

NOTATION

A	=	Area or volume where a mean value is to be estimated;
a	=	Elasticity of acceptance, when calculating the initial temperature;
b	=	Amount of dispersion around the cost of the initial solution when calculating the initial temperature;
C	=	Variogram sill;
C_0	=	Nugget effect;
D	=	Time series dimension;
F_{BEST}	=	Frequency of solutions better than the best so far;
F_{REP}	=	Frequency of accepted transitions;
h	=	Distance lag for variogram calculation
$\boldsymbol{h}$	=	Distance vector for variogram calculation;
$H(t)$	=	Marginal entropy at temperature t;
$H(x)$	=	Hessian matrix;
H_{Max}	=	Maximum entropy, equal to $\ln(I_{MAX})$;
$I(t)$	=	Number of iterations at temperature t;
$i(x,z_c)$	=	Indicator value at location x, with cut-off value z_c;
I_{MAX}	=	Maximum number of iteration at each temperature;
I_{TOTAL}	=	Total number of iterations;
k_B	=	Boltzmann constant;
L_{EXP}	=	Number of stations to be experimented (others than L_{FIXED});
L_{FIXED}	=	Number of fixed stations;
L_{TOTAL}	=	Total number of stations;
M	=	Number of points used to discritise A when kriging;
n	=	Iteration number;
OF	=	Objective function;
OF_0	=	Cost of the initial configuration;
$p_l(t)$	=	Probability that the system is in state l at temperature t;
r	=	Variogram range;
R_{STOP}	=	Number of temperature decreases without a change in the average value of the cost function;

S	=	Summation of the difference between time series;
s^2fp	=	Estimation variance obtained by the fictitious point method;
T	=	Time spent until a solution is found in SA;
t	=	Temperature in the simulated annealing algorithm;
$T(t)$	=	Time spent at iteration t;
t_0	=	Initial temperature in the simulated annealing algorithm;
TC	=	Total travel time (hours);
tc_{ij}	=	Time to travel between station i and j through the shortest path (hours);
t_{min}	=	Minimum temperature that can be attained during cooling;
UC	=	Total measuring time (hours);
uc_i	=	Measuring time in station i (hours);
V	=	Mean value of a state variable in A;
$\widehat{V}$	=	Mean estimated value of a state variable in A;
X	=	Current solution during optimization;
x_α	=	Value of the state variable at location α (known);
X_{Best}	=	Current best solution, optimal solution at the end of the annealing;
$Y_i(m)$	=	Time series value in station i, measuring time m;
$\overline{Y_i}$	=	State variable mean temporal value in station i;
Y^0_i	=	$Y_i(m)$- $\overline{Y_i}$
$Z(X)$	=	State random variable;
$z(x)$	=	Realization of a state random variable;
z_c	=	Cutoff value on variable z;
α	=	Temperature decrease factor;
$\gamma(h)$	=	Variogram;
$\gamma(h_{iA})$	=	Average variogram between the point i and the area A;
$\gamma(h_{AA})$	=	Average variogram inside area A;
Γ	=	Field where the random state variable is defined;
Δ	=	Difference in the value of the parameter;
$+\Delta$	=	Maximum allowed time shift;
$\overline{\lvert\varepsilon\rvert}$	=	Mean absolute cross-validation error;
$\lvert\varepsilon_i\rvert$	=	Absolute cross-validation error;
η_i	=	Weighting factors in the OF function;
κ_i, λ_i	=	Weights in the estimation function;
μ	=	Lagrange parameter;

σ_E^2	=	Estimation variance;
τ	=	monitoring period (*e.g.*, day, month)
$\phi(A,z_c)$	=	Spatial distribution for a random variable on a point support within region A;
$\Phi(A,z_c)$	=	Spatial random variable which generates $\phi(A,z_c)$;
χ	=	Solution space;
$\psi(A,z_c)$	=	Proportion of locations in A with cut-off value in $]z_{c-1}, z_c]$;
$\psi^*(A,z_c)$	=	Estimated proportion of locations in A with cut-off value in $]z_{c-1}, z_c]$;
ϖ	=	Cartesian of the subset of stations to be included in the new design;
Ω	=	Cartesian of the set of stations in the original design.

1 INTRODUCTION

1.1 OBJECTIVES

In this work some methods to optimize environmental monitoring networks are proposed. These methods share simulated annealing as the approximation algorithm.

A particular case of monitoring networks optimization is treated here: which stations should be included in a more cost-effective network given an existing network with a higher number of stations. Monitoring network optimization is a very actual problem given the large number of existing networks in many countries operating large numbers of stations, some of which may be redundant, with very high exploitation costs.

Difficulties appear when opposite *forces* drive the dimension of a network towards a maximum and a minimum number of stations. Exploitation costs is the driving-force pulling the dimension to a minimum and the statistical reliability of the network the driving-force pulling in the opposite direction. Finding the point of equilibrium, i.e., the optimal dimension may be a very difficult optimization problem due to the large number of combinations, even for small

network dimensions. Further complications appear when the available data is too incomplete or come from different homogeneous areas (which would ultimately lead to the need to consider each area individually). Moreover, in many cases optimal stations location is dictated by the position of a contaminant source and the predominant pathways for migration. Some practical answers to these problems were sought in this work.

1.2 JUSTIFICATION

The primary goals of monitoring programmes is to provide information to support decision making and provide the general public enough information both for environmental awareness and public participation. A critical step in developing a successful cost-effective monitoring programme is the definition of information needs. Institutions responsible for the monitoring programme management should be directly involved in the specification of information needs. Information producers and information users should first be identified, and needs should be specified according to a previous analysis of the system and available information. Information users and producers should co-operate in this regard.

The process of monitoring must be seen as a component of the monitoring and assessment cycle of environment management. Environment management, in what regards supported decision making and policy, is an activity that requires information, leading frequently to the identification of information gaps, which will induce changes or updates in existing monitoring programmes (more often updates in the monitoring network). Data collected in the existing or updated monitoring programme will again feed the decision maker with information, closing the cycle (Figure 1.1). The setting of a new monitoring programme is a process that receives information from another (or others) information cycle(s) and requires proper planning and co-operation between data producers and users to create a programme that will produce the

information identified as necessary by the parties with least financial and environmental costs (*e.g.*, the costs of network setup and costs associated with the failure to detect pollution).

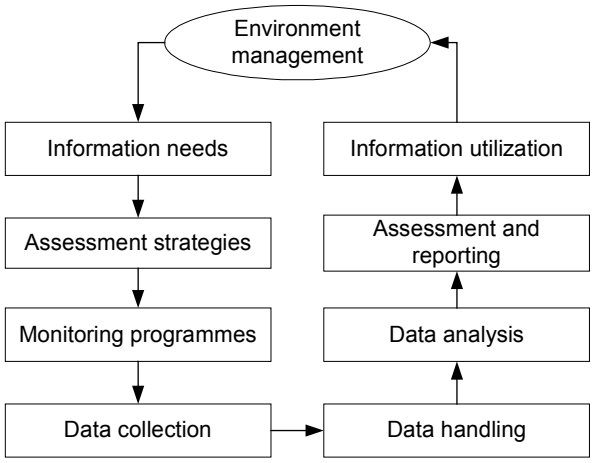


Figure 1.1. Monitoring cycle

In this work only the technical component of monitoring network design (update) is studied, and should be seen as one of the components of designing monitoring programmes. Hence it was always assumed that the remaining components in the management cycle were being considered by the programme manager (*e.g.*, a public institution) and only the geometry of the network and costs associated with it had to be included.

Environment monitoring networks optimization is a subject that has been studied for many years now. Much effort has been put in the development of statistical methodologies to design new sampling campaigns and monitoring networks (see, *e.g.*, Cressie (1991), Harmancioglu and Alspaslan (1992), Harmancioglu *et al.* (1999), Cochran (1977), and Thompson (1992)). Further developments considering also the physical reality and its variability, particularly with groundwater stochastic simulation and modelling, were the subject of works by Bras and Rodríguez-Iturbe (1976), Rouhani (1985), Loaiciga (1989), Rouhani and Hall (1988), and Pardo-Igúzquiza (1998), in the geostatistical formalisms context, Massmann and Freeze

(1987a;b), Wagner and Gorelick (1989), and Ahlfeld and Pinder (1988), in the context of geostatical simulation and modeling. These works concentrated in the optimization of a new network or on the expansion of an existing one. Network optimization when there is the need to reduce its dimension is a much less studied subject. Some examples are the works by Knopman and Voss (1989), Meyer *et al.* (1994), Reed *et al.* (2000), and Grabow *et al.* (1993). Monitoring network dimension reduction is a foreseen need in Europe and in particular in Portugal due to the already tremendous financial effort put in maintaining the actual environment network, that ascends to many thousands of stations in the European Union (EU) space, and to the need to develop specific networks imposed by EU norms. Rational allocation of financial resources will imply eliminate redundant stations from existing networks, and if needed place new ones in unsampled locations (for instance, in some European countries the already existent groundwater monitoring network density varies between a few dozen to a thousand (Scheidleder *et al.*, 2000) – the same happens in Portugal, though not included in the aforementioned work. Furthermore many of the existing networks result from aggregation of regional and local networks developed independently for specific purposes, therefore over-dimensioned to answer, *e.g.*, Waternet and Eurowaternet needs (see Nixon *et al.* (1996)). Even if the existing networks are kept operational, some representative stations will have to be chosen to make part of a specific network, which may require different monitoring frequency or specific parameters not included in the existing networks. Examples are the country-level networks for the quality of surface water, which include determination, apart from more common parameters, of heavy metals and xenobiotics (*e.g.*, Water Directive and Dangerous Substances Directive), chlorophyll (*e.g.*, Urban Waste Water Directive), ecological indicators (*e.g.*, Water Directive and 5th Environmental Action Programme). More examples and relations between water and sediment monitoring programs may be found in Nixon *et al.* (1996). Networks for soil and air monitoring need also to be implemented or rationalized. Such networks include among others, those imposed by the Landfill Directive, the Directive on the Disposal of Waste, and the Nitrate Directive, the Air Quality Framework Directive and

posterior "daughter" Directives (see EEA (1999) and EEA (2002) for more detailed information). European monitoring networks cannot be regarded as independent ones from the others because many of the measured parameters are common to many or all of them, and one environmental compartment is not separate from the others. Hence, some monitoring stations may simultaneously belong to different networks with different objectives (in the context mentioned before). Which stations from already existing national networks should be included in each of the specific EU networks? This is a question that needs not only expert judgment (as proposed by Nixon (1996)), but also optimization tools such as those developed here. Despite the particular specific technical measuring requirements needed for each particular environmental compartment, the methods proposed in this work should adapt equally well to all. The examples used are groundwater or sediment oriented, but the methods are generally applicable.

1.3 WORK ORGANIZATION

Four methodologies are proposed to handle four different practical problems when the intention is to reduce the dimension of an existing monitoring network (or prior sampling campaign): in the first two cases data is complete and full use of available information is intended; in the third case data is too incomplete and this has to be incorporated in the method; and in the fourth case data is complete, but stratification is necessary.

This thesis is organized in independent sections with their own introduction, methodology, discussion of results, conclusions and bibliography. This may lead to some redundancy particularly in the methods and references, allowing, however, the reading of each separate section without the need to cross-reference others. The organization stems from a conceptual structure similar to a compilation of individual works on a very precise theme by the same

author. This should easy the diffusion of results through the scientific community, namely by publication in international journals and at the same time reflects to the reader the background framework of the work. All the methodological and practical implementation sections have already been accepted or submitted to international scientific journals.

In the second section a bibliographic review is made. The most important works and results on the following themes are reviewed: i) methods used to design monitoring networks; ii) use of variance reduction methods applied to monitoring optimization; iii) stochastic optimization, where simulated annealing is included. This section was kept short because specific bibliographic review is made in each methodological section.

Optimization of monitoring networks is presented in sections 3 through 6. Section 3 deals with the identification of an optimal sub-set of stations given an existing set of stations with higher cardinality and complete time series. The objective is then to allocate monitoring stations considering both an optimal geographic covering of the space and eliminating stations with more redundant information (in the time series). This section presents the variance of estimation error obtained by kriging as a measure of spatial accuracy introduced in the objective function model in order to consider the spatial covering; and a new function as a measure of temporal redundancy. Methods that use variance of estimation errors are usually termed variance reduction methods. Both terms are subjected to optimization with simulated annealing. A hypothetical case-study is used to illustrate the method. An innovative component is added by the redundancy function built to use direct data values, instead of statistical transforms, as is proposed by the currently available methods.

Section 4 is a development of the first but where the costs associated with the exploitation of a groundwater monitoring network are included. Exploitation costs are divided in measuring costs, those linked directly to the time necessary to make the readings, and travel costs

associated with the time necessary to travel from station to station. The first two terms of the objective function are similar to the ones of the first section; two new cost terms are added to the objective function. Here optimization is made twice because the optimal route through the set of candidate stations is made by a third part heuristic algorithm for travelling salesman problems solution; the result of this optimization is then introduced in the objective function and subjected to optimization. Again a hypothetical case-study is used, derived from the one presented before. Innovation in this case stems from the inclusion of exploitation costs in the objective function, which has not been proposed before in the context of monitoring network optimization.

The fifth section presents a variance reduction method applied to the optimization of a real groundwater nitrate monitoring network. In this case, unlike in the hypothetical case-study of the preceding sections, time series are very incomplete. The method proposed is intended to deal with sparse data matrixes. The methods available to deal with incomplete data are based on the possibility of estimating missing data. However if data is too incomplete these methods tend to produce low quality estimates (high estimation error and variance). Monitoring network optimization is made for several measuring times selected according to a statistical criterion, resulting in an equal number of optimal networks. Selection of a final network is made by frequency analysis: only the stations that repeat more often in the optimal networks will be included in the final monitoring network.

The sixth section also presents a variance reduction method similar to the one used before applied to the selection of a small dimension sediment monitoring network given a high cardinality set of sampling stations. The objective function has only one term, the variance of estimation error, and is constrained to the reproduction of probability distribution of sediment homogeneous areas. This conditioning makes the application innovative and results in a method to select optimal spatial distribution of monitoring stations that may be considered as

a statistical stratification method without the need to define *a priori* areas of higher or lower density. The frequency of stations in each sediment homogeneous area in the resulting monitoring network is equal (or very similar) to the frequency in the original sampling campaign. This method may be used to select a set of stations to be included in a new monitoring network, or help choosing the samples to be analyzed for other physic-chemical parameters. This is useful when there are known correlation between parameters and the difference in analysis costs are high. Representative samples determined with a low cost parameter are then analyzed for other more expensive parameters.

Section 7 resumes the most important conclusions from the previous sections and integrates them in order to show the overall congruency of the proposed methods. Future developments and new research areas opened are also discussed.

All software implementations referred in this work were made in FORTRAN 90 by the author specifically to solve the problems presented and are of his full responsibility. The complete code may be found in the Appendix A..

2 BIBLIOGRAPHIC REVIEW

2.1 MONITORING NETWORK OPTIMIZATION

In this section a brief overview on environmental monitoring network optimization is made. The extent is limited by the need to keep it the less redundant with the methodological sections as possible. Hence, here an overall view of the problem is presented, underlining the most important approaches, but keeping the specific discussion to sections 3 to 6.

By definition monitoring of environmental variables comprises all sampling activities to collect and process data on some parameter (or parameters) for the purpose of obtaining information about the physical, biological and chemical properties. Monitoring practices are designed to achieve specific purposes which, in turn, lead to different types of monitoring, namely trend monitoring, biological monitoring, ecological monitoring, compliance monitoring, delineation of water quality characteristics for water use, assessment of compliance, evaluation of quality control measures, and other more specialized monitoring programs (Sanders *et al.*, 1983; Ward

and Loftis, 1986; Whitfield, 1988). The first of these types is the oldest monitoring programs and many monitoring programs have been working for several decades, though most of them were designed to answer very precise problems. Present concerns with transboundary pollution have led to the need of designing new monitoring networks, either by changing the existing ones, or by creating from scratch.

Two categories of problems arise when planning a monitoring network: i) to initiate a new network; ii) to redesign (expand or reduce an existing network). These categories will be analyzed in more detail below, but with more emphasis on redesign because it is more closely related to the objectives of the present work.

The technical design of monitoring networks is related to the determination of: i) sampling sites; ii) sampling frequencies; iii) variables to be sampled; iv) duration of sampling (the last two variables are not discussed here because they are case-specific). Most of the research results in this area have been obtained in the context of statistical procedures (Sanders *et al.*, 1983; Moss, 1986; IAHS, 1986; Whitfield, 1988; Dixon and Chiswell, 1996; Harmancioglu *et al.*, 1998; WMO, 1994). Basic statistical procedures may be found in text books by Thompson (1992) and Cochran (1977). These rely in the principle that there are several sources of uncertainty, due to measuring errors, inherent heterogeneities of the involved variables, and in the cases where modeling is involved, also simplifications and errors in both the modeling and numerical/analysis solution phase. In the context of groundwater monitoring, McBratney *et al.* (1981a), as well as many other authors after them, indicated that uncertainties are the result of lack, in quality and quantity, of information concerning the systems under study, or as a result of spatial and temporal variations of parameters. The first cause of uncertainty may be reduced by, *e.g*, increasing the number of measurements and analysis and their accuracy. The second cause of uncertainty is much more difficult to reduce, and demand for non-exact representations of the reality. Two approaches have been used in this line: i) to

model parameters as random variables with no spatial structure, resulting in monitoring plans based on random sampling (see, *e.g.*, Thompson (1992), and Cochran (1977)); ii) to model parameters as spatially distributed variables. The latter was first proposed in the context of stochastic groundwater hydrology by Freeze (1975). Since then the number of published papers on stochastic modeling and stochastic modeling/monitoring design optimization is vast (*e.g.*, Massmann and Freeze (1987a;b), Massmann *et al.* (1990), James and Gorelick (1994), and Freeze and Gorelick (1999)).

One of the most common tools used in stochastic analysis is geostatistics. In geostatistics the variable property is viewed as a spatial function with complex variations, not possible to be modeled by simple deterministic functions. These variables were coined as regionalized variables (Matheron, 1963; Matheron, 1970), and are seen as one of many possible realizations of a random field (spatial random function). More detailed descriptions of the geostatistical formalism are made in sections 3 to 6. It is worth mentioning here the three classes of methods developed for monitoring network design that are based on geostatistics: i) variance reduction; ii) simulation; iii) risk-based.

In variance reduction methods the objective is to minimize the estimation variance or some function of it, subject to constraints. These methods have been applied to the design of monitoring networks (Yfantis *et al.*, 1987; Ahmed *et al.*, 1988; Sharp, 1971; Todd *et al.*, 1976; Amorocho and Espildora, 1973; Harmancioglu and Yevjevich, 1987; Lack, 1997; McBratney *et al.*, 1981b). This method will be discussed in more detail in sections 3 to 6.

Two categories for monitoring optimization with variance reduction have been proposed: i) the local approach (*e.g.*, Amorocho and Espildora (1973)); ii) the global approach (*e.g.*, Ahmed *et al.* (1988)). In the first the influence of each additional point is analyzed separately. The variance reduction per point, VR_{o^*}, is given by (Amorocho and Espildora, 1973)

$$VR_{0*} = \frac{1}{V_*(N)}\left[C_{0*} - \sum_{i=1}^{N}\lambda_{i*}C_{i0} - \sum_{p=1}^{l}\mu_{p*}f_p(\boldsymbol{x}_0)\right]^2 \qquad (2.1)$$

where VR_{0*} is the variance reduction at $\boldsymbol{x}_0$ due to a measurement at $\boldsymbol{x}_*$; $V_*(N)$ is the estimation variance at $\boldsymbol{x}_*$ prior to the measurement; C_{0*} is the covariance function between $\boldsymbol{x}_o$ and $\boldsymbol{x}_*$; λ_{i*}, is the kriging weight of $Z(\boldsymbol{x}_i)$ in the estimation of $Z(\boldsymbol{x}_*)$ prior to the new measurement; μ_{p*} is the Lagrange multiplier for the p^{th} monomial $f_p(\boldsymbol{x}_0)$ in the polynomial drift model when estimating $Z(\boldsymbol{x}_*)$ prior to the new measurement; l is the number of basic functions; and N is the number of data points prior to the new measurement. Total variance reduction after adding one point is easily computed from the previous expression by adding the individual values at each initial location (or the points in the vicinity of $\boldsymbol{x}_*$).

In the global approach maximum or average estimation variances are used. Therefore global approaches provide only average answers to monitoring designs. It is useful to analyze designs still on the drawing board, or to perform extensive redesigns aimed to maintain the efficiency of a monitoring network, which may require removal of poorly located sites. The local approach, on the other hand, is better suited to optimally expand an existing network. The optimality, however, only relates to the additional points, which may not be acceptable if the original points are not optimal (Markus *et al.*, 1999).

In this work a global approach is used, based on the mean estimation variance obtained by averaging the kriging variance obtained by removing one of the data points and estimating it by kriging with the remaining points. A similar procedure is used in geostatistics to assess the quality of variogram models, in a routine called cross-validation (Deutsch and Journel, 1992; Journel, 1989). Other measurements of accuracy based on the estimation variance were proposed by Woldt and Bogardi (1992), namely i) the average measure of estimation variance weighted by prior suspicion, P_{ea}, and ii) the average measure of kriged contamination level weighted by estimation variance, R_a:

$$P_{ea} = \frac{1}{n_g} \sum_{i=1}^{ng} P_i \sigma_{Ei}^2 \qquad (2.2)$$

where n_g is the number of non-blank points (where information is available), P_i is the value of prior suspicion level at grid node i, σ_{Ei}^2 is the kriging variance (estimation variance).

$$R_a = \frac{1}{n_g} \sum_{i=1}^{ng} Z_i^* \sigma_{Ei}^2 \qquad (2.3)$$

with Z_i^* the expected value of contamination at point i, estimated by kriging.

When applying variance reduction methods some practical aspects must be taken into consideration, namely the type of variogram model, the nugget effect, and the screening effect. The choice of the variogram model affects the values of the kriging weights and the value of the kriging variance. Variogram models with very high continuity near the origin (*e.g.*, gaussian or power models with exponent above 1.0) produce in some cases negative kriging weights (and hence the possibility of negative estimates and negative variances); when compared with less structured models tend to give much more weight to points closer to the one being estimated (screen effect). The nugget effect, helps correcting the numerical oscillations caused by very well structured models and allows introducing variance originated by measurement errors or spatial variability in a scale inferior to the one detected by the data.

In simulation several equally probable random fields of systems' properties are generated, and physical, chemical, or physic-chemical models are run with these slightly different fields. For each run a different estimated field will result. Therefore it is possible to calculate results statistics and assess monitoring network quality.

Conditional Monte Carlo simulation techniques are particularly effective for conceptual model parameterization of highly heterogeneous media. This is due to the fact that conditional

simulation preserves the data variability but reproduces exactly (zero error) the values at the sampling locations.

Some early application of simulation with geostatistics may be found in the works of Wood and McLaughlin (1984), McLaughlin and Graham (1986), Massmann and Freeze (1987a;b), Shreve (1967). More recently, McKinney and Loucks (1992) proposed a method that use parameter measurements to minimize parameter prediction error, unlike other methods (Hoeksema and Kitadinis, 1985a; Hoeksema and Kitadinis, 1985b; McLaughlin and Graham, 1986) that use state variable measurements to minimize parameter estimation error. The problem of quantifying uncertainty in (groundwater) simulation model predictions was examined by McKinney and Loucks (1990), and recently Scheibe and Chien (2003) studied the effect of conditioning parameters estimation on the resulting flow and transport field, as measured in some monitoring wells.

Risk-based methods introduce in the definition of the problem the risk associated with missing the detection of contamination in the context of decision analysis. This risk cost, C_{risc} is included in the objective function usually in the form

$$C_{risk} = P_{fail}\mathbf{u}C_{fail} = (1-R)\mathbf{u}C_{fail} \quad (2.4)$$

where P_{fail} is the probability of failure of a proposed design configuration, **u** is the utility function, and R is the reliability. The utility function is the function that defines the level of risk aversion of a decision maker as a function of failure cost. Usually decision makers are risk averse, thus tending to weight more heavily the cost of failure as its cost increases.

The first risk-based methodology in water resources using geostatistics was introduced by Massmann and Freeze (1987a;b). The concepts had already been proposed in a series of papers by Lettenmaier and Burges (1977), Caselton and Husain (1980). More recently McKinney and Loucks (1992) compared stochastic optimization and decision analysis in the

context of aquifer remediation where location and pumping rates are to be determined, and Angulo and Tang (1999) in the context of contamination detection monitoring, in an approach that follows directly from that of Massman and Freeze (cited above).

In what regards sampling frequency, its importance may be analysed with the help of the ANOVA test, considering that step changes in the value of the variables under study. The null hypothesis, H_0, is usually that no change has occurred between two measurements; the complementary one is H_1. The probability of choosing H_0 when H_1 is true is the confidence level of the test or (1-α). The probability of choosing H_1 when H_1 is true is the power of the test denoted by (1-β).

The power of any test of effects in an ANOVA model has the general form

$$1-\beta = P\left[F_{v1,v2}(NCP) > F^{\alpha}_{v1,v2}\right] \qquad (2.5)$$

in which $F_{v1,v2}(NCP)$ is a random variable that is distributed as noncentral F with $v1$ and $v2$ degrees of freedom and centrality parameter NCP, and $F^{\alpha}_{v1,v2}$ =(1-α)th fractile point of the central F-distribution with $v1$ and $v2$ degrees of freedom. Considering J sampling stations, L sampling times, M replicates, and a null hypothesis stating that no change has occurred in our variable of interest between two times, the NCP becomes (Skalski and McKenzie, 1982; Palmer and McKenzie, 1985)

$$NCP = \left(\frac{\Delta^2 JL}{2D}\right)^2 \qquad (2.6)$$

with $v1$= 1, $v2$=2J(L-1) degrees of freedom, Δ is the overall change in the value of the variable, and D is given by

$$D = \left(\frac{\sigma_\eta^2 + \sigma_\varepsilon^2}{M}\right)$$

with σ_ε^2 the variance of the error in the observations between replicates taken at the same station on the same time, and σ_η^2 the variance of the error in the observations between sampling times at the same station.

It is clear that if the numerator in D is kept constant, increase in power is attainable by increasing either the degrees of freedom or NCP, which implies increasing the number of stations, the number of sampling times, Δ, M, some of them or all of them. As J, L and M are always integers, the problem of selecting the best combination between these parameters is also a integer (combinatorial) problem apparently much more difficult than the one treated in this work. However, for many practical monitoring networks intended to detect changes in the global state of the system under study, sampling frequency and the number of replicates may only vary in very narrow intervals due to practical reasons, and is sometimes limited to one or two values (*e.g.*, monthly or bi-monthly frequency, and three replicates). The problem is also simplified because usually frequency and number of replicates is set equal for all stations. There are nevertheless cases where tradeoffs between parameters has to be assessed. Some authors have handled this problem in different ways. The number of papers dedicated to this subject is very large and out of the scope of this work. Long reviews were made by Loaiciga *et al.* (1992), Dixon and Chiswell (1996), and in the book by Harmancioglu *et al.* (1999).

2.2 STOCHASTIC OPTIMIZATION

We concentrate in this review on stochastic methods for solving the general global optimization problem (P):

$$\begin{aligned} &\min f(x) \\ &\text{subject to } x \in S \end{aligned} \qquad (2.7)$$

where f is a function on S, and $S \subset \Re^d$ is a compact field. It is also considered that under some weak assumptions, discussed later on, the optimal solution value

$$f^* \equiv \min_{x \in S} f(x) \tag{2.8}$$

exists, *i.e*, the set $S^* \equiv \{x \in S: f(x)=f^*\}$ is nonempty and one of those solutions is attained. If f is continuous the global optimization problem (P) is inherently unsolvable in a finite number of steps (Dixon, 1978). Considering any continuous differentiable function f, any point x and any neighborhood B of x, there exists a function f' such that $f+f'$ is continuous differentiable, $f+f'$ equals f in points outside B, and the global optimum of $f+f'$ is x. Thus, for any point x, it is not possible to verify with certainty that it is not the global optimum without evaluating the function in at least one point in every neighborhood B of x. As B can be made arbitrarily small, any method designed to solve the global optimization problem would require an unbounded number of steps. Instead, one can consider that the global optimization problem is solved when a point has been found in

$$B_\varepsilon(S^*) \equiv \{x \in S : \left\| x - x^* \right\| \leq \varepsilon \text{ for some } x^* \in S\}$$

or in the level set

$$S_\varepsilon \equiv \{x \in S : f(x) \leq f^* - \varepsilon\}$$

for some $\varepsilon>0$.

This review is not exhaustive, in the sense that it does not cover all stochastic optimization methods in all their variants, but focuses on the general characteristics common to general classes of stochastic methods. Stochastic methods are understood as methods that incorporate some stochastic elements, be it in the choice of an initial solution, in the generation of candidate solutions, in the acceptance criteria, etc... It will also be seen that the concept of global optimization as defined above is not applicable (or there is no mathematical proof as to the global convergence) to practical implementations of available stochastic methods.

2.2.1 Stochastic Methods

The discussion on the available stochastic methods will concentrate on two-phase methods and random search methods. Random function approach methods are the other most important class of stochastic methods, but is not discussed here because they significantly differ from the class of methods to which simulated annealing belongs.

2.2.1.1 Two-phase methods

In two-phase methods two we can distinguish a global and a local phase. In the former phase the function is evaluated in several randomly sampled points, after which these points are manipulated (*e.g.*, by local search) to yield a candidate global optimum (minimum). The majority of two-phase methods are variants of the well-known *Multistart* algorithm, where the global phase consists of generating a sample of points from a uniform distribution over S; in the later phase a local search procedure is applied to each of the these points. The result is a set of local optima (minima).

Pure random search algorithm is the simplest global optimization method, and also an old one (see, *e.g.*, Brooks (1958 and Anderssen (1972)). The algorithm is quite simple as it consists of generating a sequence of independent identically distributed (i.i.d.) uniform points in the feasible region S, and keeping track of the best point found.

Its pseudo-code is as follows:

1. Set $n \leftarrow 1$, $y_0 \cong \propto$ (a very high value);

2. generate a point x from a uniform distribution over S;

3. if $f(x)<y_{n-1}$, then set $y_n=f(x)$ and $x_n=x$

otherwise set $y_n=y_{n-1}$ and $x_n=x_{n-1}$;

4. $n \leftarrow n+1$ and return to 2.

If f is continuous it has been proved by Brooks (1958) that the algorithm offers probabilistic asymptotic guaranty of convergence. Unfortunately the number of iterations required to find an optimal point grows exponentially with the dimension of the problem (Boender and Romeijn, 1995), which limits its applications to low dimensional problems. Other methods based in the same principle as *pure random search* include local search from a subset of the points in the sample in order to increase its efficiency. These are the *multistart*, *clustering methods*, and *multi level single linkage*.

The *multistart* algorithm incorporates local search in its simplest way by performing local search in all the points generated in the global phase. Hence better solutions, x', are searched in the neighborhood of all points x. Clearly this should improve the quality of the solutions, but has little, if any, influence in the number of iterations required to attain an optimal point (low efficiency) because it will find the same local minimum several times.

The pseudo-code is as follows:

1. Set $n \leftarrow 1$, $y_0 \cong \propto$ (a very high value);

2. generate a point x from a uniform distribution over S, and apply local search to x, resulting in x';

3. if $f(x')>y_{n-1}$, then set $y_n=f(x')$ and $x_n=x'$

otherwise set $y_n=y_{n-1}$ and $x_n=x_{n-1}$;

4. $n \leftarrow n+1$ and return to 2.

As local searches are the most time consuming part of the procedure, these should be limited to specific regions of the solution space where their contribution may be better used. *Clustering methods* and *multi level single linkage* algorithms incorporate this oriented local search by limiting it to *regions of attraction* of local maximums, x_i^*. These regions are defined as the set of points in S starting from which local search will converge to x_i^*.

Clustering methods start from a uniform sample from S (obtained in the global phase), create groups of mutually close points, and start local search no more than once in each group. Becker and Lago (1970) and Törn (1978) proposed two different ways of obtaining these groups. The first authors proposed that only a fraction of the sample with the lowest values are retained in each group (they coined it the *reduction method*); the second authors proposed that one or a few steepest descents steps from every point are performed (coined *concentration method*). In the following only the *reduction method* is considered. Two clustering rules are usually used: density clustering (Törn, 1976) and *single linkage clustering* (Boender *et al.*, 1982; Rinnooy Kan and Timmer, 1987a; Rinnooy Kan and Timmer, 1987b; Rinnooy Kan and Timmer, 1989).

The *density clustering method* (we recall that it is a variant of *clustering methods*) has the following pseudo-code (Boender and Romeijn, 1995):

1. k $\leftarrow$ 1, X*=$\varnothing$;
2. generate N points, *e.g.*, $x_{(k-1)N+1}$, ..., x_{kN}, from the uniform distribution over S, and determine the reduced sample consisting of ξ best points (with ξ a constant value) from the sample x_1, ..., x_k. Set i=1, j=1;
3. if all reduced points have been assigned to a cluster, go to 5

   otherwise

if $j \leq |X^*|$ then choose the j^{th} local minimum in X^* as the next seed point and go to 4;

if $j > |X^*|$ then apply local search to the unclustered reduced sample point $\bar{x}$ with the lowest function value, otherwise

if the resulting local minimum x^* is an element of X^* then assign $\bar{x}$ to the cluster initiated by x^* and return to 3

if $x^* \notin X^*$ then add x^* to X^* and let x^* be the next seed point;

4. add all unclustered reduced sample points which are within distance $r_i(x^*)$ of the seed point x^* to the cluster initiated by x^*

if no point has been added to the cluster for this specific value of $r_i(x^*)$ then $j \leftarrow j+1$ and return to 3

else increment $i \leftarrow i+1$ and repeat 4;

5. $k \leftarrow k+1$ and return to 2.

With continuous f, Rinnooy Kan and Timmer (1987a;b) proposed that $r_i(x^*)$ be equal to

$$r_i(x) = \frac{1}{\sqrt{\pi}} \left(i \cdot \Gamma(1+\frac{d}{2}) \cdot \sqrt{-\det H(x)} \cdot m(S) \cdot \frac{\zeta \ln(kN)}{kN} \right)^{\frac{1}{d}} \quad (2.9)$$

where $H(x)$ is the Hessian matrix at x and ζ is a positive constant.

In the *single linkage clustering* clusters are formed sequentially. After a cluster, C, is initiated by a seed point, an unclustered point, x, is found such that

$$d(x,C) = \min_{y \in C} \|x - y\| \quad (2.10)$$

is minimal. The point is added to C. The process continues until $d(x,C)$ exceeds some critical value r_k. The pseudo-code is as follows (Rinnooy Kan and Timmer, 1987a; Rinnooy Kan and Timmer, 1987b):

1. $k \leftarrow 1$, $X^* = \varnothing$;

2. generate N points, *e.g.*, $x_{(k-1)N+1}$, ..., x_{kN}, from the uniform distribution over S, and determine the reduced sample consisting of the γkN best points from the sample x_1, ..., x_k. Set j=1;
3. if all reduced points have been assigned to a cluster, go to 5
4. otherwise
5. if $j \leq |X^*|$ then choose the j^{th} local minimum in X^* as the next seed point and go to 4;
6. if $j > |X^*|$ then select as the next seed point the unclustered reduced sample point $\bar{x}$ with the lowest function value. Apply local search to $\bar{x}$ to find a minimum x^* and add x^* to X^*;
7. add all unclustered reduced sample points which are within distance r_k of a point already in the cluster initiated by the seed point x selected in 2 to the cluster. $j \leftarrow j+1$and return to 3;
8. $k \leftarrow k+1$ and return to 2.

Rinnooy Kan and Timmer (1987a;b) proposed that r_k be equal to

$$r_k = \frac{1}{\sqrt{\pi}}\left(\Gamma(1+\frac{d}{2})\cdot m(S)\cdot\frac{\zeta \ln(kN)}{kN}\right)^{\frac{1}{d}} \qquad (2.11)$$

Despite the theoretical guaranty with probability one of a finite number of local searches, which proves the efficiency of the method, there is no asymptotic guaranty of convergence to the optimal value. Moreover it can be shown that some local minima may be missed (Boender and Romeijn, 1995). *Multi level single linkage* deals with this problem, combining the efficiency of single linkage with the theoretical convergence guaranty of *multistart*. In *multi level single linkage* local search is applied to every sample point, except if there is another sample point within some critical distance with a smaller function value. The pseudo-code is as follows (Rinnooy Kan and Timmer, 1987a; Rinnooy Kan and Timmer, 1987b):

1. $k \leftarrow 1$, $X^*=\varnothing$;

2. generate N points, *e.g.*, $x_{(k-1)N+1}$, ..., x_{kN}, from the uniform distribution over S, and set i=1;
3. if there exists some j such that $f(x_j)<f(x_i)$ and $\|x_j - x_i\| < r_k$ then go to 4, otherwise apply local search to x_i, and add local maximum found to X^*;
4. $i \leftarrow i+1$. If $i \leq kN$ then go to 3, otherwise $k \leftarrow k+1$, and go to 2.

The critical distance is determined again by equation (2.11).

The aforementioned methods aim at finding all local optima values of optimization problems. However this approach may not be the best strategy when the number of local optima is very large (which is very often the case). Other methods must be used if one wants to look for the global optimum. In the following *random search methods* will be presented and their theoretical convergence to global optimality discussed.

2.2.1.2 Random search methods

In this class of methods are included those that generate a sequence of points in the feasible region following some prespecified probability distribution or sequences of probability distributions. Two main alternatives exist: i) points are generated from a single probability distribution – points are iid variables – ; and ii) points are generated from an adaptative probability distribution, which is depends on the iteration number and on previous iteration points. Three algorithm classes are discussed: i) *random search*; ii) *pure adaptive search*; iii) *adaptive search*.

Random search is in its essence similar to the pure random search method mentioned earlier about two phase methods. However, here the uniform distribution is replaced by any

distribution whose support has nonempty interception with S^*. This substitution does not affect the probabilistic asymptotic guaranty of optimal convergence (when the sample size grows to infinity) (Boender and Romeijn, 1995). The number of points is also dependent on the iteration number and need not be decided *a priori* (class of algorithms introduced by Solis and Wets (1981)). The pseudo-code for this algorithm, considering $\{\mu_n\}_{n=0}^{\infty}$ a sequence of probability distributions on $\Re^d$:

1. Set n=0 and choose $x_0 \in S$;
2. generate y_{n+1} from μ_n;
3. set $x_{n+1} = D(x_n, y_{n+1})$, $n \leftarrow n+1$ and return to 2.

The map D, S x $\Re^d$, and range S satisfies the condition

$$f(D(x,y)) \leq f(x) \text{ and if } y \in S, f(D(x,y)) \leq f(y)$$

Solis and Wets (1981) proved that the sequence $\{(f(X_n)\}_{n=0}^{\infty}$ is monotone convergent, decreasing in this case (increasing in the authors' original proof), and that the sequence converges to the optimum with probability one as n grows to infinity.

If improvement is forced at each iteration then the algorithm becomes the pure adaptive search, as introduced by Patel *et al.* (1988) and Zabinsky and Smith (1992). In this algorithm an iteration point is generated from the uniform distribution on the subset of points that are improving with respect to the previous iteration point:

1. Set n=0 and $y_0=\infty$;
2. generate x_{n+1} uniformly distributed in $S_{n+1}=\{x \in S: f(x)<y_n\}$;
3. set $y_{n+1}=f(x_{n+1})$, $n \leftarrow n+1$and return to 2.

Zabinsky and Smith (1992) proved that for global optimization problems over a convex feasible region and continuous distribution function of the variable given by $U=f(X)-f^*$, when X has the uniform distribution on S, the number of iteration grows linearly with the dimension of the problem. This is an interesting result because it points to the possibility of finding an efficient random search method for global optimization. Despite this nice property, two main difficulties arise when implementing pure adaptive search methods (Boender and Romeijn, 1995): i) constructing the improving region $S_n=\{x \in S: f(x)<f(x_{n-1})\}$, and generating a point uniformly distributed in S_n. To overcome this difficulty Rubinstein (1981) proposed that points should be generated from a Boltzmann distribution, π_t, with density function given by

$$g_T(x) \propto e^{f(x)/t} \qquad (2.12)$$

with t is some small positive value termed temperature. Romeijn and Smith, 1994) showed that for all $\varepsilon>0$, $\lim_{t\downarrow 0} \pi t(S_\varepsilon) = 1$. The pseudo-code with these changes is the following:

1. Set $n=0$, $t_0=\infty$, $y_0=\infty$;
2. generate x from the distribution π_{t_n} over S,

   if $f(x)<y_n$, set $x_{n+1}=x$

   otherwise repeat 1
3. set $y_{n+1}=f(x_{n+1})$ and set $t_{n+1}=\alpha(y_{n+1})$, $n \leftarrow n+1$ and return to 2

Here the parameter t_n changes with the iteration, or as a result of the value attained by the function, and its value is determined by some nonincreasing $\Re$ function, α. Instead of sampling from a nested set of smaller level sets, in this algorithm sampling is done from the feasible region S. Pure adaptive search may be considered as a particular case of adaptive search: in the case when $t_n=\infty$ for all n, then all points have equal probability of being sampled. Romeijn and Smith (1993) showed that convergence to the global optimum is attained with stochastically less iterations with adaptive search than with pure adaptive search. Once again,

the problem of generating points according to a specified distribution arises. It is necessary to find a way to generate points according to the Boltzmann distribution for $t>0$. Some authors have proposed that instead of trying to generate points from a distribution π exactly, Markov chains be constructed with easily generated transitions, and with limiting distribution equal to π (Smith, 1984; Berbee *et al.*, 1987; Bélisle *et al.*, 1993; Boender *et al.*, 1991; Deyer and Frieze, 1992). The most well known adaptive search type algorithm (approximate algorithm, in the sense that distribution π is approximated) is Simulated Annealing (SA).

2.2.2 Simulated annealing

The discussion on simulated annealing convergence properties is given here along the lines of Romeijn and Smith (1994), Dekkers and Aarts (1991), Bélisle (1992), Aarts and Korst (1990), Otten and van Ginneken (1989), and on the review by Boender and Romeijn (1995), and discussed more in detail because one SA implementation was built in this work to obtain high quality solutions to minimization problems. Practical aspects of its implementations are discussed in latter sections; in this section attention is concentrated on more theoretical considerations.

As mentioned earlier, it is very difficult to generate points directly from the distribution π. Therefore SA uses an approximating approach based on a random walk on S, which converges to the uniform distribution on S. If the transition probability given that the Markov chain is in state $x \in S$ is denoted by $R(x,.)$; and the random walk is filtered such that in every iteration, given iteration point x_n, a point z_{n+1} is generated from $R(x,.)$. Then, the point is accepted with probability

$$\mathrm{P}\{\text{accept } z_{n+1}\} = \begin{cases} 1 & \text{if } f(z_{n+1}) \le f(x_n) \\ \exp\left[\dfrac{f(x_n) - f(z_{n+1})}{t_n}\right] & \text{if } f(z_{n+1}) > f(x_n) \end{cases} \quad (2.13)$$

which is the Metropolis criterion (Metropolis *et al.*, 1953). By filtering the Markov chain by this transition probability (R), the resulting sequence of points converges to the Boltzmann distribution π_t (Boender and Romeijn, 1995). Hence it is possible to generate a sequence of points $\{X_n(t)\}_{n=0}^{\infty}$ with the property that for every ε>0

$$\lim_{n\to\infty} P(X_n(t) \in S_\varepsilon) = \pi_t(S_\varepsilon) \quad (2.14)$$

and by recalling that $\lim_{t\downarrow 0} \pi_t(S_\varepsilon) = 1$ from the discussion on adaptive search, conclude that

$$\lim_{t\downarrow 0} \lim_{n\to\infty} P(X_n(t) \in S_\varepsilon) = 1 \quad (2.15)$$

Simulated annealing makes use of this result by not only applying the transition probability *R*, but also by making the parameter *t* converge to 0 by using an adaptive cooling schedule, α (the reason why the decrease in *t* is called cooling will be discussed latter).

The general SA pseudo-code is as follows:

1. Set n=0, choose $t_0 \in [0,\infty]$, $x_0 \in S$;
2. select y_{n+1} according to the transition probability distribution $R(x,.)$;
3. if $f(x_{n+1}) \le f(y_{n+1})$, set $x_{n+1} = y_{n+1}$. If $f(y_{n+1}) > f(x_{n+1})$, set $x_{n+1}=y_{n+1}$ with probability given by equation (2.13); otherwise set $x_{n+1}=x_n$;
4. set $t_{n+1}=\alpha_{n+1}(x_0,\ldots,x_{n+1})$, $n \leftarrow n+1$ and return to 2.

It is clear that SA will accept moves that deteriorate the quality of the solution, but in a limited way: at high *t* values most of the transitions are accepted; while at low *t* values essentially improving transitions are accepted.

Equation (2.15) states the asymptotic condition of convergence to global optimality of SA. However, there are not many SA algorithms for which global convergence has been proved. The only two exceptions are the algorithms of Romeijn and Smith (1994), Romeijn and Smith (1993) and of Dekkers and Aarts (1991) developed to very particular problems. In general SA behaves as an optimization algorithm only if it is allowed an infinite number of transitions. Approximating the asymptotic behavior arbitrarily closely requires a number of transitions that for most problems is typically larger than the size of the solution space, leading to exponential-time execution of the algorithm. SA is therefore unsuited to solve optimization problems to optimality, but the asymptotic behavior can be approximated in polynomial time.

2.2.2.1 Finite-time approximation

We will concentrate now in finite-time approximation to SA as it is the basis for the development of efficient algorithms, even if one knows that this efficiency means only attaining a good quality solutions (eventually not the optimum) in a feasible amount of time, *i.e.*, the time necessary to obtain the solution, up to some prescribed error, grows only polynomially with the dimension of the problem. One may ask why using an algorithm that is known not to give the optimal solution when there are many methods that have been proved to converge to global optimality. Two reasons may justify such choice: i) either the other methods take too much time to compute the optimal solution, and a faster, though not as accurate method, is needed; or ii) there is no other method available to solve the problem. Most combinatorial problems belong to ii).

Finite-time implementations of SA can be realized by generating homogeneous Markov chains of finite length for a finite sequence of descending values of the control parameter t (cooling). Other parameters must also be determined. The set of parameters that control the cooling and

the dimension of the Markov chains are combined in a so-called cooling schedule, and include an initial value of the parameter t, t_0; a decrement function of t, a final value of t, specified in a stopping criterion; a finite length of each homogeneous Markov chain (homogeneous meaning that the transition rule is equal). More evolved SA implementations have dynamic decrement functions and lengths of the Markov chain (*i.e.*, they are not necessarily always equal).

The control parameter t_n and the length of the Markov chains L_n are the two principal parameters when discussing cooling schedules. This is so because they are used to control the approximation to the stationary statistical distribution, π_{t_n} at each t_n.

We introduce now the concept of quasi equilibrium as defined by Aarts and Korst (1990). Quasi equilibrium is achieved if the probability distribution of the solutions after L_n trials of the n^{th} Markov chain, $\kappa(L_n,t_n)$, is sufficiently close to the stationary distribution, π_{t_n}, or

$$\left\| \kappa(L_n, t_n) - \pi_{t_n} \right\| < \varepsilon$$

The length of the Markov chains and the decrement function must be chosen such that quasi equilibrium is restored at the end of each individual Markov chain. Clearly, if the decrement is large, the length of the Markov chain will have to be longer to restore equilibrium. There is a tradeoff between these two parameters, and either the decrements are made very small to keep L_n small (the most common approach), or decrements are large and L_n very large (less common). Quasi equilibrium and parameters tradeoff are central on the search for adequate cooling schedules. In a cooling schedule choices have to be made about: i) the initial value of t, t_0; ii) the temperature function, $\alpha(x,n)$; iii) the number of iterations to be performed at each t_n, i.e., L_n; iv) a stopping criterion to terminate the algorithm.

A great variety of cooling schedules have now been suggested. The earliest proposed cooling schedules were based on physical analogy. For instance, Kirkpatrick *et al.* (1983) set t_0 to a

very high value in order to guaranty that all transitions were accepted at the beginning of the run, and used a geometric cooling function, *i.e.*, $t_{n+1}=\alpha t_n$. L_n is determined by a sufficient number of transitions being accepted subject to an upper bound. Other simplifications on this scheme may set L_n constant, or dependent on the size of the problem instance or neighborhood size. If the objective function is not altered for a number of t decreases, the algorithm is stopped.

In Aarts and Laarhoven (1985), Laarhoven and Aarts (1987) and Aarts and Korst (1990), t_0 is calculated such that approximately all transitions are accepted, but unlike the empirical value set by Kirkpatrick *et al.* (1983) they propose that m_0 initial trials be performed, requiring that the initial acceptance ratio $\chi_0=\chi(t_0)$ is close to 1, where $\chi(t)$ is defined as the ratio between the number of accepted transitions and the number of proposed transitions. The initial value of t is then obtained from the following expression

$$t_0 = \overline{\Delta f^+}\left[\ln\frac{m_2}{m_2\chi_0 + (1-\chi_0)m_1}\right]^{-1} \qquad (2.16)$$

where m_1 and m_2 denote the number of trials ($m_1+m_2=m_0$) with $\Delta f_{xy}\leq 0$ and $\Delta f_{xy}>0$, respectively, and $\overline{\Delta f^+}$ the average value of the latter values ($\Delta f_{xy}=f(x)-f(y)$). The decrement function is given by

$$t_{n+1} = t_n\left[1+\frac{t_n\ln(1+\delta)}{3\sigma(t_n)}\right]^{-1} \qquad (2.17)$$

where $\sigma(t_n)$ denotes the standard deviation of the values of the cost function of the points in the Markov chain at t_n, and is a small positive value. The constant δ is called the distance parameter and determines the speed of the decrement of the control parameter. The stopping criterion is given by

$$\left|\frac{\mathrm{d}\bar{f}_s(t_n)}{\mathrm{d}c}\frac{t_n}{\bar{f}(t_0)}\right| < \varepsilon_s \qquad (2.18)$$

where $\bar{f}(t_0)$ is the mean value of the points found in the initial Markov chain, $\bar{f}_s(t_n)$ is the smoothed value of $\bar{f}$ over a number of chains in order to reduce the fluctuations of $\bar{f}(t_n)$, and ε_s is a small positive real value. This stop criterion is based on the idea that the average function value of a Markov chain is an increasing function of t, such that it converges to $f(x_{\min})$ as $c \downarrow 0$. The length of the Markov chain is kept constant and equal to $L=L_0 \cdot \iota$, where ι denotes the dimension of S and L_0 a constant called standard length. Extensions of these schedules to continuous functions were made latter by Dekkers and Aarts (1991) when proving global convergence of simulated annealing in continuous functions optimization.

Despite of the convergence results that resulted in the previous cooling schemes, that are only two examples from a already large set of alternatives (see, *e.g.*, Collins *et al.* (1988), Hajek (1988)) showed that for asymptotic convergence for the set of optimal solutions, the condition is for the cooling to be carried out sufficiently slow, rather than there necessarily being any requirement to attain equilibrium at each t_n. Also, if t is kept fixed for a number of iterations, there is a tradeoff between large reductions in t and a small number of iterations at fixed t, as seen before. This results led to the development of cooling schedules that, in the extreme cases, include a single iteration at each temperature, with the following alternative functions (Lundy and Mees, 1986)

$$t_{n+1} = \frac{t_n}{1 + B \cdot t_n} \text{ and } t_n = \frac{C_1}{1 + C_2 \cdot t_n} \qquad (2.19)$$

where B, C_1 and C_2 are constants. L_n and stopping criterion are set as with the algorithm proposed by Kirkpatrick *et al.* (1983). Another alternative is the fixed cooling schedule (Connolly, 1988; Fielding, 2000) that proposes a fixed t_n for all or almost all iterations.

We mention also the logarithmic schedule proposed by Hajek (1988), in the form

$$t_n = \frac{c}{\log(n + n_0)} \qquad (2.20)$$

where n_0 is a positive integer (*e.g.*, 1) and c is the smallest value for which an SA with this cooling schedule is convergent, and $c \geq d^*$, where d^* is the largest depth of the local minima, which are not global minima. This scheme was proved by Bruce Hajek to be the fastest logarithmic-type cooling scheme leading to convergence.

Though there exist theoretical proof of convergence for some of these schedules, their implementation has led to either poor solutions when the time is limited, or to extremely large running times. Some authors have proposed changes to the basic SA algorithm in order to improve performance. One alternative is to store the best solution during run (see, *e.g.*, Glover (1964)). Another alternative is to sample the neighborhood without replacement, *i.e.*, when the T approaches 0 the new solution is picked-up from the set of neighbors and it is guaranteed that all neighborhood is searched (see, *e.g.*, Johnson *et al.* (1991a;b)). Two different search strategies have been used: either the neighbors are selected at random, or in a sequential way. None seems to outperform the other. The advantage of this approach is that when the algorithm has been terminated and there has been no improvement in the solution for at least b iterations (where b is the size of the neighborhood), then the final solution is guaranteed to be a local minimum.

One of the most promising SA implementations is its parallelization, for which there have been several interesting results for quite a long time now (Aarts and Korst, 1990; Durand and White, 2000; Onbasoglu and Ozdamar, 2001). Its isn't however clear yet if parallelization will work well for most of the practical cases.

New and improved versions of simulated annealing have been proposed in the last years, of which the most promising seems to be the Adaptive Simulated Annealing (ASA) proposed by Ingber (1989), in which the cooling schedule is controlled by sensitivity coefficients (similar to those obtained in gradient methods) and parameters are allowed to vary at each iteration

inside an interval that is also dependent on the iteration. Some recent applications have been reported in bibliography with good results (see, *e.g.*, Chandler *et al.* (1999), Ingber and Wilson (2000), and Chen *et al.* (2001)), though still more thorough bench testing are needed before assuming ASA as a state of the art algorithm in any application.

3 GROUNDWATER MONITORING NETWORKS OPTIMIZATION WITH REDUNDANCY REDUCTION

3.1 SYLLABUS

Three optimization models are proposed to select the best subset of stations from a large groundwater monitoring network: i) one that maximizes spatial accuracy; ii) one that minimizes temporal redundancy; and iii) a model that both maximizes spatial accuracy and minimizes temporal redundancy. The proposed optimization models are solved with simulated annealing, along with an algorithm parameterization using statistical entropy. A synthetic case-study with 32 stations is used to compare results of the proposed models when a subset of 17 stations are to be chosen. The first model tends to distribute the stations evenly in space; the second model clusters stations in areas of higher temporal variability; and results of the third model provide a compromise between the first two, *i.e.*, spatial distributions that are less regular in space, but also less clustered. The inclusion of both temporal and spatial

information in the optimization model, as embodied in the third model, contributes to selection of the most relevant stations.

3.2 INTRODUCTION

Monitoring network dimension reduction is a need foreseen in Europe and other developed countries due to the already substantial financial effort put into maintaining the present environmental monitoring network. This amounts to many thousands of stations in the European Union (EU) space and the need to develop specific networks imposed by EU norms (see Nixon *et al.* (1996) for a discussion on this theme). Rational allocation of financial resources means eliminating redundant stations from existing networks and placing new ones in unsampled locations, where necessary. Which stations from already existing national networks should be included in each of the specific EU networks? This is a question that needs not only expert judgment (as proposed by Nixon (1996)), but also optimization tools such as those presented herein.

The optimization of environmental monitoring networks is a subject that has been studied for many years. Much effort has been put into the development of statistical methodologies to design new sampling campaigns and monitoring networks (see, *e.g.*, Harmancioglu and Alspaslan (1992), Harmancioglu *et al.* (1999), Thompson (1992)). Further developments considering the physical reality and its variability, particularly with groundwater stochastic simulation and modeling, were examined by Massmann and Freeze (1987a;b), Wagner and Gorelick (1989), Ahlfeld and Pinder (1988) in the context of geostatistical simulation and modeling. These works concentrated either on the optimization of a new network or on expansion of an existing one. Network optimization when there is the need to reduce its dimension has been much less studied. Exceptions are the works by Knopman and Voss (1989), Meyer *et al.* (1994), Reed *et al.* (2000), and Grabow *et al.* (1993).

Monitoring networks for areal mean rainfall events was the subject of early works by Rodríguez-Iturbe and Mejía, (1974a;b), where they considered the spatial and temporal variability of mean rainfall. The variance of mean rainfall was calculated as the product of point process variance, a reduction factor due to sampling in time (dependent only on the correlation in time and length of the time series), and a reduction factor due to sampling in space (dependent on the spatial correlation structure, the sampling geometry and the number of stations). The authors studied random and stratified random sampling schemes, and obtained abacus for different correlation functions, number of stations, and area of the region. Lenton and Rodríguez-Iturbe (1974) further considered the density and location of the stations. Bras and Rodríguez-Iturbe (1975) compared sampling schemes for different point variances, covariance functions and covariance functions parameter. Bras and Rodríguez-Iturbe (1976) included in the same formalism the cost associated with each station to help choose the best set (*i.e*, number and position) of stations with less cost and less mean rainfall variance. Delhomme (1978) applied the geostatistical fictitious point method (usually used to assess the quality of covariance models when estimating with kriging) to determine the optimal location of rain gauges. If the number of stations is large, then the dimension of the combinatorial problem may be exhaustively intractable. Pardo-Igúzquiza (1998) solved this problem with a metaheuristc approach (simulated annealing). Rouhani (1985) also used variance reduction techniques to determine the number and position of groundwater monitoring stations, and also analyzed the robustness and resilience of the variance reduction analysis. Rouhani and Hall (1988) proposed the incorporation of risk, defined as the weighted sum of the expected value and the estimation variance, in order to correct the "blindness" of the estimation variance to extreme values. The method also considers temporal changes in the hydrologic variables. Loaiciga (1989) also proposed a variance reduction method using time-dependent spatial models (based on space-time means and covariances), with good results. This author used the mixed integer programming model of Hsu and Yeh (1989), originally developed for optimum design for parameter identification. Maximum periodicity was allowed

in this study, but did not include trade-off analysis between sampling periodicity and further increasing the number of stations. Later developments in the solution of time-dependent models were proposed by Pardo-Igúzquiza (1998), with the inclusion of the climatological variogram (Angulo and Tang, 1999; Lebel *et al.*, 1987). More extensive reviews on this topic may be found in Loaiciga *et al.* (1992), Dixon and Chiswell (1996), and Harmancioglu *et al.* (1999).

Variance reduction techniques use the variance of the estimation error (σ_E^2) as an indicator of the accuracy of the estimated values. In geostatistics, the mathematical definition of σ_E^2 means that its value does not depend on the actual values of the measured variables, but on the relative spatial distribution of the measuring locations. Therefore, one may use σ_E^2 as an indicator of which spatial distribution is best for a sampling network by testing all the combinations between available sampling locations and selecting the combination that minimizes σ_E^2.

Temporal redundancy is considered to be a function of the *similarity* between time series. By adopting this approach, stations that show larger differences between them over time and simultaneously have the *best* spatial distribution are retained. In principle, this would generate a network of monitoring stations smaller than the original, but which best reproduces the temporal and spatial behavior of the state variable. Temporal redundancy reduction had already been proposed by Amorocho and Espildora (1973), Caselton and Husain (1980), Harmancioglu and Yevjevich (1987), Husain (1989), and Harmancioglu and Alspaslan (1992). Harmancioglu *et al.* (1999) reviewed its applications in the context of information theory. Despite the elegance of this method, it is limited by the need to assume a probability distribution for the variables, which may be unknown or difficult to determine. Moreover the

method is particularly well adapted to variables with equal probability distributions (usually Normal or lognormal).

Space-time models have already been applied by, *e.g.*, Angulo and Tang (1999), Lebel *et al.* (1987), Buxton and Pate (1994), Dimitrakopoulos and Luo (1994), and Pardo-Igúzquiza (1998). These approaches require complex variogram fitting and are intended to reproduce the primary space-time nested variances. Pardo-Igúzquiza (1998) also used simulated annealing to solve the same problem with good results in a network augmentation problem.

Considered herein is the problem of reducing the dimension of an existing network of groundwater monitoring (GMN) stations from Ω to ω, $\omega<\Omega$. Both the spatial covariance and available time series of a state variable are used. The problem is solved by replacing one station at a time and evaluating the effect on the cost function. Simulated annealing (SA) is used to approximate the solution. It should also be emphasized that the algorithms usually referenced as SA are in reality simulated quenching because the temperature schedule used is exponential instead of logarithmic (see Ingber (1993) for a discussion). Faster cooling schedules are necessary to circumvent the slow convergence rate of SA, and have been proved to give good results, but without guaranteeing to find the global optimum. The exponential schedule is used herein and is referred to as SA in order to maintain the current nomenclature. Three different models for the optimization model are proposed and tested: i) one considering only spatial information in the form of accuracy of spatial estimates; ii) one that considers only temporal information, by reducing redundancy in the time series; and iii) a model that includes both space and time information.

3.3 SPACE AND TIME MODELS FOR NETWORK OPTIMIZATION

Some groundwater monitoring programs have, in the past, been implemented on the basis of spatial and temporal intensive monitoring programs. These have shown to require extensive manipulation or expensive and sensitive hardware. Cost rationalization and the need to orient monitoring programs to other specific objectives have made it necessary to reduce the size of the existing monitoring networks. The new size and spatial distribution must, however, be sufficient to ensure that both spatial and temporal variabilities are correctly included in the new design. In many instances, some of the monitoring stations must be retained in the future design. This is the case for wells used to produce tap water, to detect specific contamination sources, or because their position is considered strategic for any other reason. These constraints are easily implemented in the optimization algorithms if known *a priori*. A direct comparison method between all time series is used in this study that considers all the available raw data. A new definition for data temporal redundancy is proposed and used to obtain better spatial distribution of monitoring stations. Temporal redundancy and a measure of spatial coverage quality (*i.e*, variance of the error of estimation obtained by kriging, σ_E^2) are combined in a single model and subjected to minimization. Simulated annealing (SA) is used to find the solution with entropy used to help parameterize the SA algorithm.

3.3.1 Estimation variance

Field data within the geostatistical formalism is considered to be the result of random processes of regionalized variables, *i.e.*, of random variables with space coordinates, with some spatial covariance. Regionalized variables are continuous in space, and therefore not completely random, but at the same time it is not possible to model them by a deterministic

function (or spatial process.) They therefore lie between deterministic and stochastic processes (Matheron, 1970), thus incorporating the notion of uncertainty in the conception of inference models or in the simulation of variables (Matheron, 1970; David, 1977; Journel and Huigbrejts, 1978). A review of the use of geostatistics for mapping and sampling design appears in ASCE (1990a;b).

The values of a state variable $z(x)$ at the sampled points in the field can be considered as realizations of a set of random variables $Z(x)$ in a field Γ. A set of random variables $Z(x_i)$ defined in a field Γ is a random function $Z(\boldsymbol{x})$. Some restrictions with respect to stationarity are needed. The most common theory considers that the distribution function is invariant by translation and intrinsic stationarity. If increments are made at step h, then the resulting expression is the variogram:

$$\gamma(\boldsymbol{h}) = \frac{1}{2} E\left\{\left[Z(\boldsymbol{x}+\boldsymbol{h}) - Z(\boldsymbol{x})\right]^2\right\} \tag{3.1}$$

It is a necessary and sufficient condition that the variogram is a conditionally negative definite function.

If an estimate of the mean value of a state variable in an area A from values at locations x_α, $Z(x_\alpha)$, inside or outside the area is $\hat{V}$, then a linear estimation of $\hat{V}$ can be obtained from p data points by

$$\hat{V} = \sum_{i=1}^{p} \kappa_i \cdot Z(x_i) \tag{3.2}$$

which is unbiased if the sum of the weights κ is one. This is a common requirement in several methods, and also in kriging. The kriging system is (Journel and Huijbregts, 1978):

$$\begin{cases} \sum_{i=1}^{n} \kappa_i \cdot \gamma(h\ ) + \mu = \gamma(h_{iA}) \\ \sum_{i=1}^{n} \kappa_i = 1 \end{cases} \tag{3.3}$$

where μ is the Lagrange parameter, and $\gamma(h_{iA})$ is the average variogram between the point i and the area A when one extreme of the vector h is fixed in x_i and the other extreme describes the area A independently. The optimal estimation variance value is expressed by (Journel and Huijbregts, 1978),

$$\sigma_E^2 = \sum_{i=1}^{n} \kappa_i \cdot \gamma(h_{iA}) - \gamma(h_{AA}) + \mu \tag{3.4}$$

with $\gamma(h_{AA})$ being the average variogram inside A. The estimation variance is a measure of the estimation accuracy of V. Because σ_E^2 only depends on the geometric configuration of the data points, and, once a variogram model is defined, it is possible to change data locations and calculate the estimation variance again. The spatial arrangement of points that minimizes σ_E^2 has the lowest estimation error, and therefore best reflects the spatial correlation introduced in the variogram model. By minimizing σ_E^2 the stations that can best be predicted by the remaining stations are excluded. The estimation variance is incorporated in the objective function model in a reduced form:

$$\sigma_E^2(n) = \frac{(\sigma_E^2(n))}{Max(\sigma_E^2)} \tag{3.5}$$

where n is the iteration number. In order to have a value that varies between fixed boundaries, [0,1], σ_E^2 is divided at each iteration by the maximum estimation variance found so far. The estimation variance is calculated at each iteration (combination of stations) using p (=ω) stations.

3.3.2 Redundancy

Time series are considered to be synchronous, either because data are collected at the same time, or because the necessary interpolations are made to synchronize data. Time events can therefore be handled as realizations of random functions, $Y_i(m)$, $i=1,\ldots, L_{FIXED} + L_{EXP}$; $m= 1,\ldots,D$, (with L_{FIXED} being the number of fixed stations, *i.e.*, those that are to be included in all solutions; L_{EXP} the dimension of the subset of stations to be included in the new design, and D the dimension of the time series vectors). For the sake of simplicity, time indices are used only when needed; mathematical operations with time are made over all times. $Y_i(m) \in X \in \chi$, with X being the current solution, χ the set of possible solutions, and i = 1,…,ω. The sum of the differences between time series, S, is used to evaluate temporal redundancy: large values of S indicate that series are significantly different, while smaller values indicate the opposite. The sum S is an approximation to the sum of the integrals between time series, or to the sum of the areas defined between any two time functions. In order to calculate temporal redundancy, time series should have an equal mean value, with redundancy depending only on the time series variances and temporal fluctuations. Hence, a new variable is defined, $Y_i^0(m) = Y_i(m) - \overline{Y}_i$, with $\overline{Y}_i$ being the mean temporal value of the state variable in station i. The summation of the difference between series is made for all time periods, with the possibility of shifting one time series in relation to the others by the time value d so that the sum is the lowest, *i.e.*, when summing the difference between series $Y_i^0(m)$ and series $Y_k^0(m)$ $k \neq i$, only the minimum summed values are used. Minimum temporal redundancy means maximum S:

$$S(n) = Max\left\{ \sum_{\substack{i=1 \\ k=i+1}}^{L_{FIXED}+L_{EXP}-1} Min\left\{ \frac{\sum_{m=1}^{D-d} Y_i^0(m) - Y_k^0(m+d)}{D-d}, 0 \le d \le +\Delta \right\} \right\} \qquad (3.6)$$

where $+\Delta$ is the maximum allowed time shift. $S(n)$ is incorporated in the objective function model in a reduced form: $S(n)$ is kept between fixed bounds,[0,1], by dividing it by the largest $Max(S)$ found so far:

$$S(n) = \frac{S(n)}{Max(S)} \tag{3.7}$$

n is the iteration number.

This approach is believed to outperform statistical methods because i) extreme values with low frequency tend to be masked by most statistical methods, ii) seasonality and trends are difficult to handle and pre-processing is usually required to filter these features; iii) statistical methods that handle seasonal phenomena well (*i.e*, either in correlation or frequency domains) are ill suited to handling trends; iv) non-parametric methods well suited to handling complex data *(e.g*., factorial analysis) require human intervention and are therefore not appropriate for automatic procedures.

3.3.3 Resulting optimization model

Consider the following optimization model:

$$\min \delta \cdot \sigma_E^2(n) + \eta\left[-(S(n)\right] \tag{3.8}$$

where δ *and* η are weighting factors that allow the spatial component to be weighted differently from the temporal component in the objective function.

Three different optimization models are compared by: i) setting η equal to zero, and therefore including only the spatial component in the objective function; ii) setting δ equal to zero, and therefore including only the time component; iii) setting both δ *and* η equal to one, thereby building a space-time model with equal weighting. Different weightings may be used to compensate for different statistical reliability. The problem is single-objective as it depends only on the selected set of stations.

The problems presented in this section and in section 4 include formulations of the objective functions with two or more terms dependent on the same variable, the set of stations. The problem was chosen to be solved as a single objective one, considering the terms as equally important. A multiobjective approach might also have been used. In this case the set of non-dominated solutions (Pareto-optimal solutions) had to be found and an appropriate solution chosen according to the decision maker's judgment of the relative importance of each term.

3.4 SOLVING THE OPTIMIZATION PROBLEM

3.4.1 Iterative improvement

The problem of reducing a monitoring network of dimension Ω to a network of smaller dimension ω requires testing all possible combinations of ω in Ω. The number of possible combinations is given by $\Omega!/(\omega \cdot (\Omega - \omega)!)$. If the dimension of the network is large, the number of combinations may be high. The proposed solution to the combinatorial problem involves replacing one station at a time, evaluating the result, retaining the station if it reduces the objective function, or if the result fulfills a probabilistic criterion (the Metropolis criterion (Metropolis *et al.*, 1953)), and rejecting the station otherwise. The iterative process of replacing the stations and analyzing the fulfillment of the Metropolis criterion is known as simulated annealing. The algorithm examines at each iteration the cost of a given set of stations (solution), and it may happen that a station that is rejected in one iteration, with a particular combination of stations, is accepted latter if the combination is different (different solution). It is guaranteed that most of the solution space is searched. SA is one of the threshold algorithms included in the class of local search algorithms. The other two, as defined by Aarts and Korst (1997), are: iterative improvement, where only objective function-reducing neighbors are accepted; and threshold accepting, where some deterministic non-increasing threshold

sequence is used. In the latter method, neighbor solutions with larger objective function values are accepted, but to a limited extent, because the threshold value is fixed and always decreasing, with a very rigid control on the size of the cost difference. Simulated annealing uses a more flexible control on the values of the threshold, allowing transitions out of a local minimum. SA was first introduced by Kirkpatrick *et al.* (1983) as an algorithm to solve well-known combinatorial optimization problems that reduces the risk of falling into local minima (or metastable solutions) common to iterative improvement methods. This is a consequence of SA accepting only solutions that lower the objective function. These authors proposed the use of the Metropolis (Metropolis *et al.,* 1953) procedure from statistical mechanics, which generalizes iterative improvement by incorporating controlled uphill steps (to worse solutions). The procedure states the following: consider one small random change in the system at a certain temperature; the change in the objective function is ΔOF; if $\Delta \text{OF} \leq 0$, then the change in the system is accepted and the new configuration is used as the starting point in the next step; if $\Delta \text{OF} > 0$ then the probability that the change is accepted is determined by $P(\Delta \text{OF}) = exp(-\Delta \text{OF}/k_b T)$; a random number uniformly distributed in the interval (0,1) is taken and compared with the former probability; if this number is lower than $P(\Delta \text{OF})$ then the change is accepted. The control parameters $k_b T$ are replaced by the parameter t (also called temperature) to avoid using the Boltzmann constant, k_b, which would have no meaning in the present context.

The SA algorithm operates in the following way: i) the system is *melted* at a high temperature (initial temperature, t_0); ii) the temperature is decreased gradually until the system *freezes* (*i.e*, no further objective function change occurs); and iii) at each iteration the Metropolis procedure is applied. The following describes how to establish the initial temperature, the rate of temperature decrease, the number of iterations at each temperature, and the stopping criteria to halt the process when the system is *frozen*. The generic SA algorithm incorporates: a neighborhood function $N()$, a heuristic h_N, a random function $P()$, the iterations counter c, and

the current temperature t: $c \leftarrow 0$; $t \leftarrow t_0$. Choose a feasible solution $X \in \chi$, $X_{best} \leftarrow X$; Do while $c < c_{MAX}$, or $t > t_{min}$

$$
\begin{array}{l}
\text{do}\ \Big\{ \begin{array}{l}
U \leftarrow h_N(X) \\
\text{if } U \neq Fail \\
\text{then}\ \Big\{ \begin{array}{l}
\text{if } P(U) > P(X) \\
\quad \text{then}\ \Big\{ \begin{array}{l} X \leftarrow U \\ \text{if } P(X) > P(X_{best}) \\ \quad \text{then } X_{best} \leftarrow X \end{array} \\
\text{else}\ \Big\{ \begin{array}{l} r \leftarrow Random(0,1) \\ \text{if } r < e^{[P(U)-P(X)]/t} \\ \quad \text{then } X \leftarrow U \end{array}
\end{array} \\
c \leftarrow c+1 \\
t \leftarrow \alpha \cdot t
\end{array} \\
\text{Return } (X_{best})
\end{array}
$$

Figure 3.1. Simulated annealing algorithm. *Random* is a random real number taken from a uniform distribution in (0,1).

In order to accelerate the process, several improvements have been proposed for limiting the number of iterations at each temperature. The dimension of the Markov chain (in a Markov chain the probability of the outcome of a given trial depends only on the outcome of the previous trial, as occurs in SA) has been proposed to be a function of the dimension of the problem (Kirkpatrick *et al.,* 1983). Temperature is maintained until 100Ω solutions (iterations), or 10Ω successful solutions have been tested, whichever comes first, where Ω is the number of variables (stations) in a problem. These authors propose that the annealing process cease if, after three consecutive temperatures, the number of acceptances is not achieved. The annealing can also be stopped if the average value of the objective function does not change after a pre-established number of temperature decreases (R_{STOP}). Along with these dynamic criteria, a static one may be used to halt the process when a minimum temperature, t_{min}, is attained. The former guarantees that the annealing will stop if none of the dynamic

criteria are fulfilled, even before the total number of iterations is attained. In the algorithm used herein, both the dynamic and the static criteria are implemented.

Cunha and Sousa (1999) proposed the following expression to calculate the initial temperature, t_0,

$$t_0 = -\frac{b \cdot OF_0}{\ln a} \tag{3.9}$$

where OF_0 is the cost of the initial configuration, a is the elasticity of acceptance, and b is the amount of dispersion around the cost of the initial solution. A prior model run determines the latter. The elasticity of acceptance represents the probability of accepting a solution worse than the initial one. The initial temperature determined by equation (3.9) is such that there is a probability a of accepting solutions that are b% worse than the initial solution. This approach is similar to the one proposed by Cunha and Sousa (1999), but here the amount of worsening is established. Temperature is usually decreased with a constant rate (cooling factor) α such that after s temperature decreases, the temperature is $t_s=t_0 \cdot \alpha^s$. The two stopping criteria: t_{min} and number of temperature decreases are complementary because it is easy to calculate the minimum temperature attained if t_0 and α are known. Criterion t_{min} may be useful if one wants to stop the annealing after a certain temperature is reached.

The generation mechanism of a new solution, U, from the neighborhood, $N(X)$ of a solution X is a simple one: Consider the solution vector $X=\{x_1, \ldots, x_\omega\}$; i) randomly remove one element (station) from X; ii) randomly choose an element in $\chi \setminus X$; iii) insert this element in X, obtaining U.

3.4.2 Analysis of the convergence

Due to the similarity between simulated annealing and the physical equivalent, a measure of disorder similar to entropy may be used to assess the evolution of the annealing process.

Entropy is a classical statistical mechanics definition of a thermodynamic system used to evaluate the state of disorder. Marginal entropy is given by (if the Boltzmann constant is set equal to one and natural logarithms are used):

$$H(t) = -\sum_{l \in \chi} p_l(t) \ln p_l(t) \tag{3.10}$$

where $p_l(t)$ is the probability that the system is in state l at temperature t, and χ is the solution space. The summation is carried out over the states allowed by the system. As temperatures decline, the number of states found that improve the best so far should also decrease, which is equivalent to the system reaching a more organized structure, or that entropy is decreasing. Marginal entropy should decrease to ln $|\chi|$ if equilibrium is attained at each temperature (Aarts and Korst, 1990). If the best solution is unique, then marginal entropy converges to zero. It can be seen that the value of $H(t)$ is maximum when all states have equal probability of occurrence, and that $H_{max}(t) = \ln (I_{MAX})$. I_{MAX} is the dimension of the Markov chain at each temperature (*e.g.*, 100Ω or 10Ω).

Relative entropy is defined as

$$\frac{H(t)}{H_{\max}(t)} = \frac{H(t)}{\ln(I_{MAX})} \tag{3.11}$$

Relative entropy should decrease from a sill (≈ 1) as temperature decreases, attaining at the limit and the optimum a value close to zero. Hence, at sufficiently low temperatures the probability of finding better states tends to zero. If the algorithm is operating well, and if the tuning parameters are correct, then this should happen. This was used as a criterion to select the best set of annealing parameters. Entropy therefore seems to be an appropriate indicator of how well the search is made, and whether it is converging to viable solutions. Although it does not indicate if the solution found is a global one, it can help to find good parameters for the simulated annealing algorithm. A good convergence schedule must fulfill the following criteria: i) the time spent at high temperatures must be long enough to allow running through many of the candidate solutions, *i.e.*, the variables must oscillate for the first high

temperatures, with the exception of H/H_{max}, which must be close to one; ii) as temperature decreases, S must converge to its maximum, σ_E^2, and the objective function must converge to their minima, and H/H_{max} must converge to $\ln|\chi|$; and iii) for sufficiently low temperatures, no oscillation is expected. These criteria have been elegantly described in the theory of simulated annealing by Aarts and Korst (1990), and correspond to two well-defined regions in the convergence curve: region 1 and region 2. In region 1 the objective function values are distributed uniformly (modeled with a normal density curve) and the structure of the combinatorial problem plays a minor role; in region 2, close to the minimum, the structure of the combinatorial problem determines the solution density (as modeled with other density curves). Similar conclusions are also drawn from the entropy curves: at high temperatures the system is disordered and the entropy is maximum, as compatible with a normal density curve; at low temperatures entropy is minimum and should be modeled with densities that minimize entropy (*e.g.*, exponential). Entropy analysis should help in the selection of a good set of SA parameters if the criteria mentioned previously are fulfilled, although it does not indicate the values directly.

3.5 APPLICATION EXAMPLE

To test the three models, a grid of 32 GMN stations randomly distributed in a square grid of eight by eight spatial units was created (Figure 3.5). Monitoring stations numbers 1 and 2 represent wells for water supply, and are therefore be included in all the solutions. Hypothetical time series for all stations are calculated using common mathematical functions, both purely deterministic and with normally distributed errors. The functions were selected based on empirical judgment and experience, and attempt to mimic the behavior of water quality variables. Some of the resulting variables are non-homocedastic (*i.e.,* variance is

dependent on the magnitude of the value), thus reflecting a feature common to variables such as redox potential and electrical conductivity when the scale of measure has to be changed between consecutive observations. With the method proposed here, no statistical assumption on the data is made, which accelerates the pre-processing stage. The time series equations are presented in Table 3.1, where τ stands for unitary measuring interval (*e.g.*, one day, one months), $k_1\tau$ stands for the first record moment and $k_n\tau$ for the last, k= 1, 2, ….

Table 3.1. Time series equations

Series number	**Equation: x =**
1	$Cos(k\tau)+0.54\ k\tau$
2	$0.12\ k\tau$
3	$Sin\ (k\tau)$
4	$Cos((k+1)\ \tau)$
5	$Cos((2k+1)\ \tau)$
6	$Log(k\tau) - Log(k_n\text{-}k_1)$
7	$Exp[-k\tau\ /\ (k_n\text{-}k_1)]$
8	1
9	$Cos(0.3\ k\tau + 3\ (k_n\text{-}k_1))$
10	$Atan(k\tau)$
11	$Atan(k\tau + 1.5\ (k_n\text{-}k_1))$
12-22	Repeat 1-11 + 20% normally distributed error around the mean
23-32	Repeat 1-10 + 40% normally distributed error around the mean

It is now intended to choose the 15 monitoring stations that should be added to stations 1 and 2 in order to obtain a network with lower exploitation costs (considered dependent on the number of stations) that best preserves the temporal features of the data. This is the network that eliminates stations which add less information and has a *good* spatial distribution, given a variogram model. It is assumed that the costs of sampling are much lower than those of the chemical analysis, and that the number of stations was set such that they can all be sampled in a single day. Therefore the cost of sampling is only dependent on transport costs. In this case Ω is equal to 30 and ω is equal to 15, and the problem amounts to finding the minimum objective function value within a solution space of more than 1.55×10^8 possible combinations.

3.6 RESULTS AND DISCUSSION

It should be noticed now that the example state variable of the case-study is generic in that it represents the general behavior of natural parameters such as water levels, concentration of a chemical element, temperature, etc. Therefore the values of S and σ_E^2 are given in *unit* (un) and *square unit* (un^2), respectively; objective function and H/H_{max} are dimensionless.

In order to guarantee stationarity of the Markov chains, a sufficient length of time must be spent at each temperature. Moreover, the amount of time spent at high temperatures must also be large enough to allow running through many of the candidate solutions, to avoid being trapped in local minima. The questions to be answered are: i) how high should the initial temperature be; ii) how many iterations are to be made at each temperature; and iii) how fast should the temperature fall. If the search is to take a feasible amount of time some other aspects must also be considered: i) the number of iterations at each temperature has to be the lowest possible; ii) the number of temperature reductions should be optimized; and iii) the search must stop if a *crystallized* system is attained.

Relative entropy may be used to complement the analysis of the change in the objective function, σ_E^2, and S as temperature changes. What should be examined is the value of t_0 for which the objective function is minimized, and such that relative entropy is kept at high values for the first few iterations, decreasing to zero after that. This would indicate that the probability of accepting a new solution decreases as temperature decreases, starting with a value near one, where most of the solutions are accepted and the probability of accepting worse solutions decreases as crystallization is attained.

The time spent by the search at high temperatures should be large enough to avoid being trapped in local optimal solutions, but small enough to minimize running time. Equation (3.9) allows the time spent at the higher temperatures to be tuned, and also enables the amount of dispersion around the objective function value of the initial solution to be set. Both parameters were evaluated in prior batch runs, as described subsequently. The effect of adding temporal information to the optimization of the monitoring network is studied by comparing the resultant network when using only spatial information, only temporal information, and using both.

By construction, the objective function may decrease if a maximum of S or σ_E^2 is found in any iteration. This is due to division by the maximum value whereby the objective function attains relatively small differences in equal solutions (since different maxims are used), but random jumps out of local minima are also allowed. These jumps are less frequent for lower temperatures because a larger part of the solution space was searched. Even if they occur, the probability of missing the neighboring minimum decreases with t because most of the allowed transitions become more strictly descending.

An experimental spherical isotropic variogram for the first time period was fitted with the following parameters: nugget = 8.1×10^{-3} un²; sill = 8.01×10^{-2} un²; range = 2.482 m (a more detailed analysis is presented in section 3.8). A specific computer program was developed (OPTIVAR) and run on Pentium III (Intel 800 MHz) computers. Some prior batch runs using equation (3.9) were undertaken to obtain the proper initial temperature. The results indicated an initial temperature of 1.424, and this was used in all subsequent runs. Convergence was evaluated by analyzing the evolution in the variables S, σ_E^2, objective function, processing time, and H/H_{max}, with temperature.

3.6.1 Network optimization only with spatial information

Both σ_E^2 and objective function decreased from high values to their minima (Figure 3.2a and b). At high temperatures the objective function and σ_E^2 show high variability, indicating that worse solutions are still being accepted, but this variability decreases with temperature. No variability was detected at low temperatures, indicating that no further improvements were possible. The system unable to jump to worse solutions because the probability of drawing a random number that fulfills the Metropolis criterion is low, as discussed previously. Marginal entropy also fulfills the criteria for convergence with a long sill for high temperatures. It decreases for lower temperatures, converging to zero as the best solution is found (Figure 3.2c). The minimum objective function value is attained at 0.3294, corresponding to a σ_E^2 of 4.244 $x10^{-2}$ square units. The average processing time was 993 seconds.

When only spatial information is used, the objective function is only dependent on σ_E^2 and the sampling stations must therefore be distributed in space, thereby minimizing the variance of the estimation error. Table 3.2 shows the two sets that result for 50 runs, along with the frequency of occurrence of the resulting solution sets. The two solution sets have equal final objective function values, and are therefore equally *good*. Both solutions are shown in Figure 3.3. The minimum objective function value is different from the minimum σ_E^2 because the cost is obtained after the division of σ_E^2 by its maximum value (equation (3.10)).

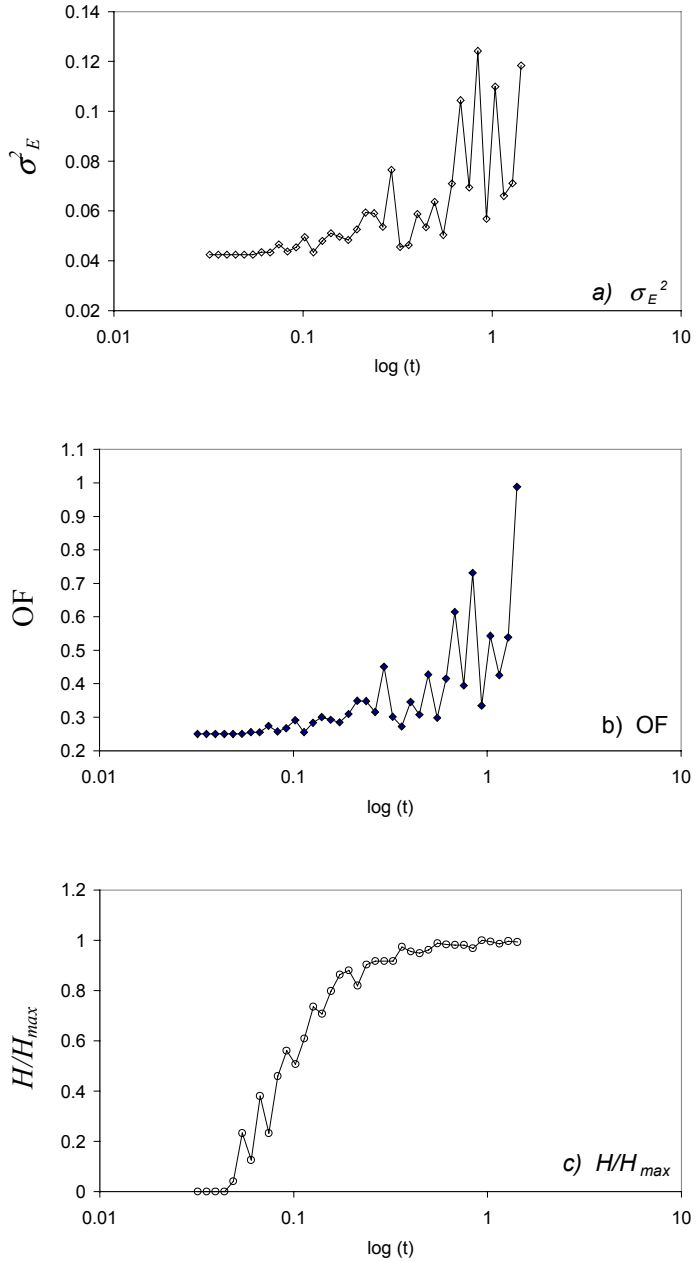


Figure 3.2. Minimizing only σ_E^2 : a) σ_E^2 (un²)*;* b) OF; c) H/H_{max}

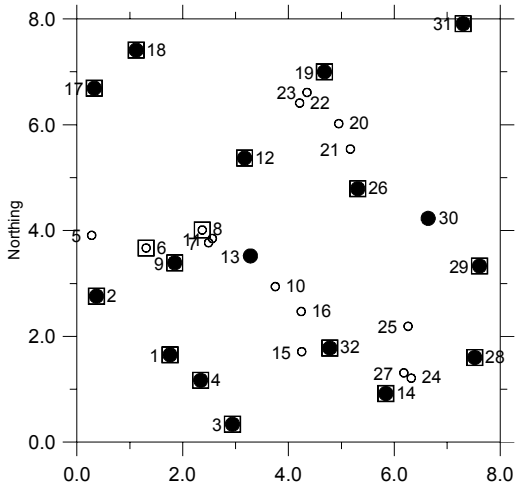


Figure 3.3. Minimizing only σ_E^2 : optimal solution. Solution A1 in squares and solution A2 in filled circles

Considering only the spatial component, the stations are uniformly distributed in space. Similar results were obtained by Sacks and Schiller (1988) and Pardo-Igúzquiza (1998) for spatial models (without sampling costs in the case of the latter). The two solution sets (A1 and A2) have similar spatial distribution and correspond to geometric configurations with the same estimation variance. Had an anisotropic variogram been used, the spatial distribution would be different. One interesting feature of the data used in this case-study is that variogram models are similar in all time periods. In cases where the variogram models vary with time care is needed when identifying the period(s) when the spatial variability is best reproduced, and this/these is/are not necessarily when it is higher. High spatial variabilities may occur because of particular events, like extraordinary pumping regimes, and thus do not reflect the natural state of the system.

The frequency of occurrence of the two solutions is similar (55.6 % and 44.4 %, for A1 and A2, respectively). The difference between the two solutions are stations 6, 8, 13 and 30 (in bold in Table 3.2). The problem is therefore non-unique and some other criterion has to be used in choosing one over the other, such as travel costs and sampling costs.

Table 3.2. Solutions: Minimizing only σ_E^2, maximizing only *S*, and minimizing σ_E^2 and maximizing *S*. OF: objective function

Solution Name	Solution Set	Frequency %	σ^2_E (un²)	*S* (un)	OF
Only spatial information					
Solution A1	{1,2,3,4,**6**,**8**,9,12,14,17,18,19,26,28,29,31,32}	55.6%	4.244×10^{-2}	-	0.3294
Solution A2	{1,2,3,4,9,12,**13**,14,17,18,19,26,28,29,**30**,31,32}	44.4%	4.244×10^{-2}	-	0.3294
Only temporal information					
Solution B	{1,2,3,19,20,21,22,23,24,25,26,27,28,29,30,31,32}	100%	-	438.2	6.96×10^{-8}
Both spatial and temporal information					
Solution C1	{1,2,3,**12**,**16**,**17**,19,20,21,22,25,27,28,29,30,31,32}	87.5	8.07×10^{-2}	335.7	0.557
Solution C2	{1,2,3,**8**,**18**,19,20,21,22,**23**,**26**,27,28,29,30,31,32}	12.5	8.09×10^{-2}	335.8	0.596

3.6.2 Network optimization with only temporal information

When only temporal information is used, the objective function dependents solely on the value of *S*. The objective function should converge to zero because the spatial component of the objective function model was set constant and equal to zero; as *S* is obtained by the division by the maximum *S* value found so far, the temporal component of the objective function should converge to zero. The objective function value obtained was 6.96×10^{-8} (Table 3.2). The discrepancy between this value and that expected (zero) may be justified by numerical oscillations and truncations during data storage and handling. The solution was attained at *S* = 438.24. It is clear from Figure 3.4a and b that both *S* and objective function converge to a maximum and a minimum, respectively. Relative entropy also shows a good convergence curve, fulfilling the convergence criteria stated in section 3 (Figure 3.4c). The resulting solution is shown in Table 3.2 as solution set B, and is represented in Figure 3.5. The average processing time was 886.5 seconds.

If only temporal information is used then the spatial distribution of the stations is dictated by local variability of the time series and by their mutual redundancy. The spatial coverage was more clustered than when using only spatial information, which may be explained by the need to choose stations that are close by but which have significantly different time series, as

happens with stations 19 to 32 (with high random fluctuations around the mean). In practical problems this result may be a consequence of having stations near faults, pumping areas, contaminations sources, or any other source of strong local variability.

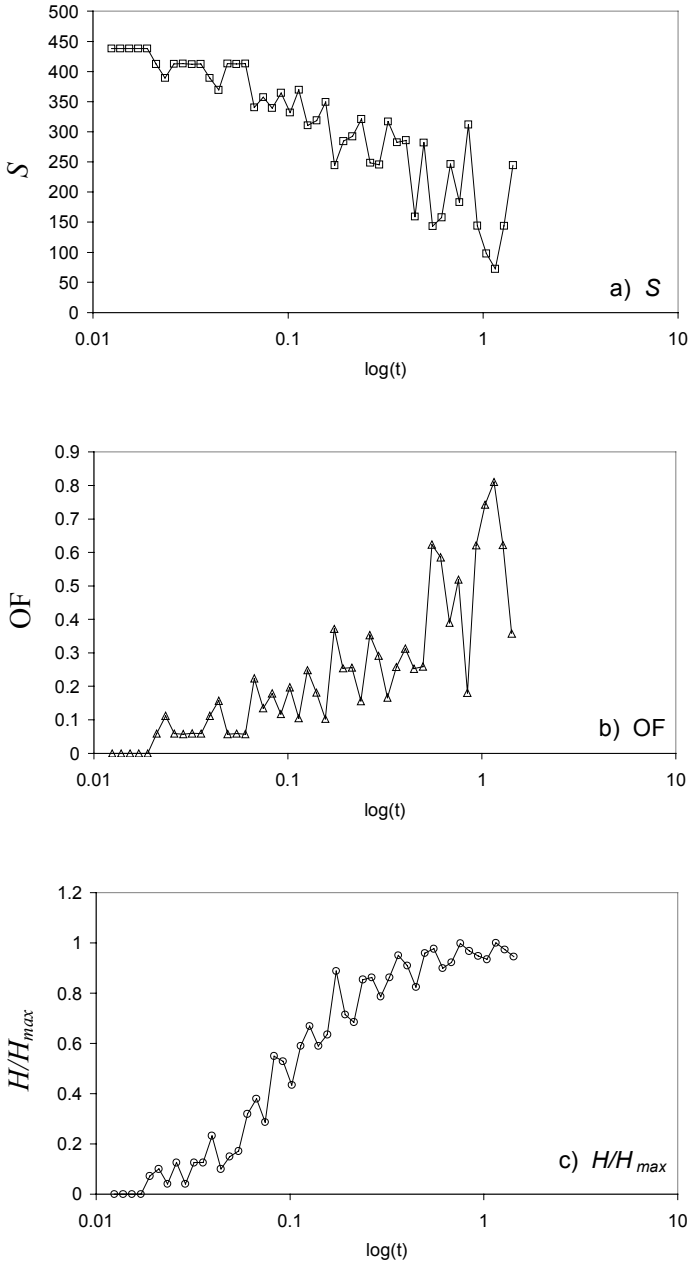


Figure 3.4. Maximizing only S: a) S (un); b) OF; c) H/H_{max}.

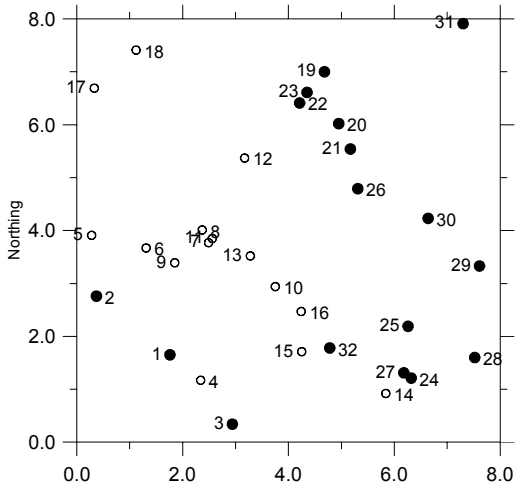


Figure 3.5. Maximizing only *S*: optimal solution. Solution B is shown in filled circles.

3.6.3 Space-time network optimization

If both spatial and temporal information is considered, then the spatial distribution of stations is expected to be between the regular coverage proposed by the estimation variance and the more clustered distribution proposed by *S*. Again, the convergence curves fulfill the convergence criteria (Figure 3.6a and b). The minimum objective function value was attained at 0.557 for σ_E^2 = 8.07x10^{-2} squared units and S = 335.7. As expected, the solution for the complete model is between those for the single components: σ_E^2 is now higher and S is now lower. Objective function value is now 40% higher than it would be if the components were independent (summing the objective function values for the first two models: objective function = 0.399), and the resulting solution sets (C1 and C2, see Table 3.2) are different from solution sets A and B, and could not be obtained by any simple combination of the latter. The frequency of the two solution sets is very different (87.5 % and 12.5 %, for C1 and C2, respectively) and does not correspond to equally good solutions – the costs are different. Rather, it indicates that a worse solution, very near to the best, was found one in every eight runs. Unlike the results of

the first model, the best solution in this case is known, and no other criterion is needed to choose between them. The average processing time was 1012 seconds.

The resulting monitoring network (Figure 3.7) has the advantage of incorporating both spatial and temporal information, thereby maximizing the quality of the data collected. The space component of the model is reflected in a network with higher spatial accuracy, and the temporal component simultaneously selects the stations that are less redundant in time.

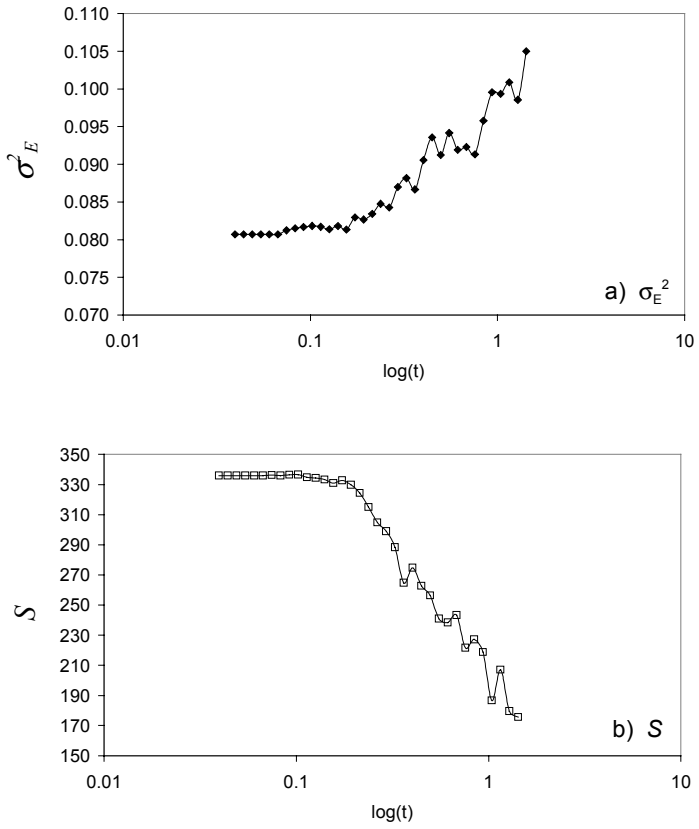


Figure 3.6. Minimizing σ_E^2 and maximizing S: a) σ_E^2 (un²); *b)* S (un); c) OF; d) H/H_{max}

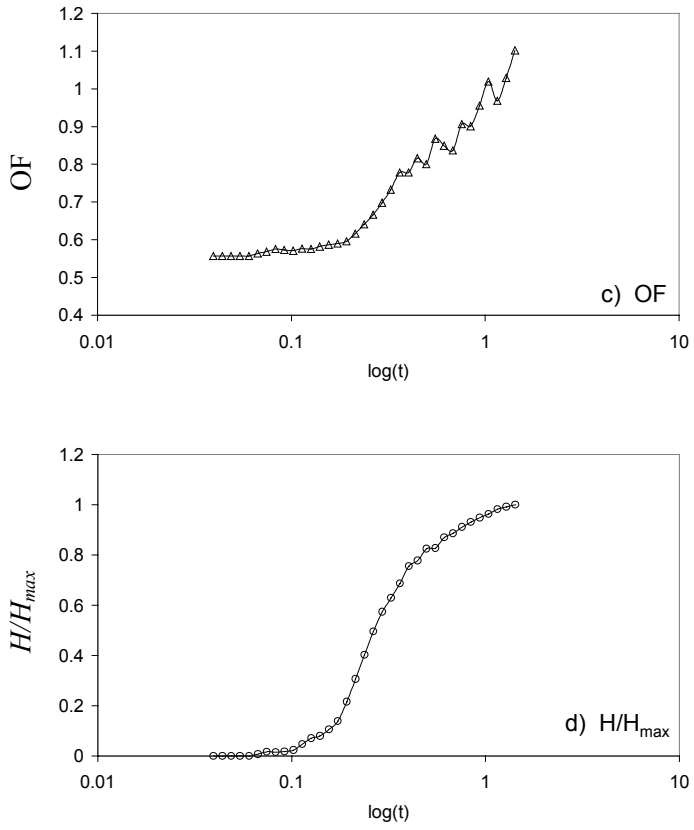


Figure 3.6. Minimizing σ_E^2 and maximizing S: a) σ_E^2 (un²); *b)* S (un); c) OF; d) H/H_{max} (cont.)

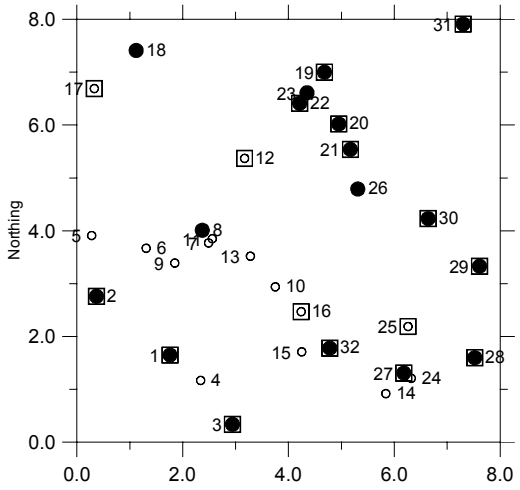


Figure 3.7. Minimizing σ_E^2 and maximizing S: optimal solution. Solution set C1 is represented in empty squares and solution set C2 in filled circles

3.7 CONCLUSIONS

Three different models for the determination of the best set of monitoring stations of a monitoring network were evaluated. The models were designed to accommodate problems of network reduction when time series are available. This is usually the case in regional or sub-regional (*e.g.*, aquifer scale) monitoring networks that have resulted from the aggregation of several local monitoring plans, and that must be transformed in a smaller network which still captures the regional variability of the state variable. A synthetic case study was used, and simulated annealing was selected as the algorithm to solve the combinatorial problem.

A method based on the statistical entropy of simulated annealing chains (Markov chains) was proposed to help choosing the SA parameters, and applied to the case-study. The results showed that it is a good tool.

The first model considered only a measure of the quality of the spatial distribution, in the form of kriging variance. Results indicated that with this model the best spatial distribution is a uniform one (the stations are evenly distributed in space), as given by an isotropic variogram model.

The second model used a new variable, the temporal redundancy of the time series, to select the set of stations that had minimum common information, or that collectively assembled most of the variability of the time series possible with the reduced network. The temporal redundancy variable proposed in this article requires no assumption as to the probability distribution because it uses the sampled values directly, rather than a statistical transform. The results indicated that the spatial distribution may be more clustered than the one obtained with an optimization model with only a spatial component, if the time series less

correlated in time are spatially clustered. This may be the case in areas of high heterogeneity, such as near important faults, at contacts between different flow regimes, or near pumping areas.

The third model included both the spatial and the temporal components. In the method proposed here only space variography is required, and therefore the pre-processing time is greatly reduced when compared to other approaches cited in "Space-time network optimization". The results of the space-time model show that the spatial distribution of monitoring stations with space-time information is less regular than that obtained with only spatial information, and less clustered than that resulting from time information only.

The space-time model proposed for selecting the subset of stations from a monitoring network best reflects the spatial variability of the state variable, and simultaneously includes the least redundant stations. The monitoring network thus obtained maximizes the relevance of the data collected, and contributes to a better cost-benefit ratio. This model seems a good alternative to other methods proposed in the literature.

3.8 ANNEX TO SECTION

Eighteen (k_1=1 and k_n=18) values were generated for each time series with a constant value of τ, equal to one. Time series were then converted to the same order of magnitude (equivalent to converting to the same unit of measurement), and then centralized by dividing by their mean value. Structural analysis was made for the first period and considered equal for the remaining periods. A problem where different variogram models are considered is studied in section 5. The experimental variogram is shown in Figure 3.8 along with the theoretic

spherical variogram model, manually adjusted to the data, given by the following analytical expression

$$\gamma(h) = C_0 + \begin{cases} C \cdot \left[1.5\frac{h}{r} - 0.5\left(\frac{h}{r}\right)^3 \right] & if\ h \le r \\ C & if\ h \ge r \end{cases} \qquad (3.12)$$

where C_0 is the small scale variability due to analytical errors or to variability associated with structures with dimension inferior to the spacing between data locations, also called nugget effect; C is the amount of correlated variability, also called sill; r is the distance after which there is no evident spatial autocorrelation (the experimental variogram is flat or oscillates around a constant value), also called range of the variogram.

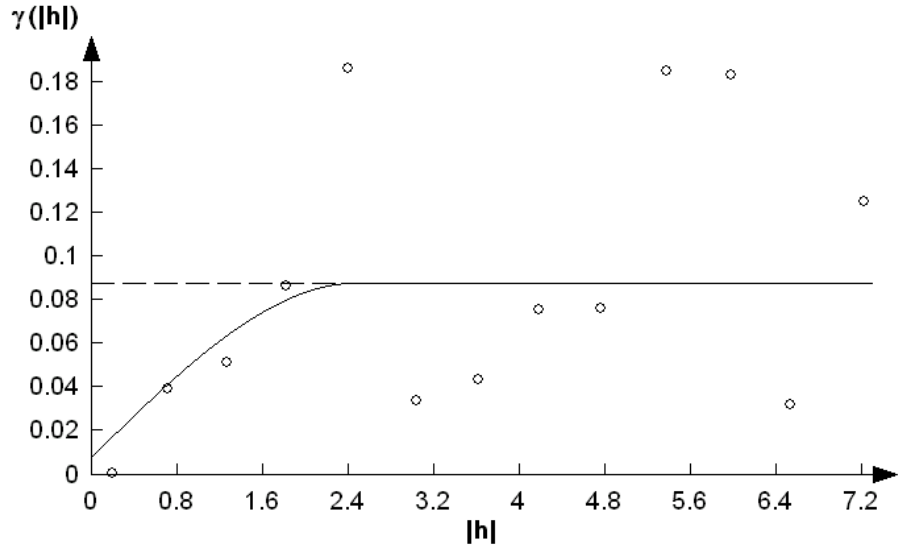


Figure 3.8. Experimental omnidirectional variogram and adjusted spherical model

Directional experimental variograms were too erratic to allow any conclusion about anisotropy. This is a result of the presence of different time series functions and to the fact that they were constructed independently (with no imposed covariance). Hence, a single

omnidirectional variogram was adjusted. Data statistics and the resulting variogram model are shown in Table 3.3.

Table 3.3 Data statistics and spherical variogram parameters

Ω	$\overline{m}$ **(un)**	s^2 **(un²)**	C_0	C	r
32	1.076×10^{-1}	9.017×10^{-2}	1.007×10^{-2}	8.01×10^{-2}	2.482

4 OPTIMAL SPACE-TIME COVERAGE AND EXPLOITATION COSTS IN GROUNDWATER MONITORING NETWORKS

4.1 SYLLABUS

A method to determine the optimal subset of stations from a reference level groundwater monitoring network is proposed. The method considers the redundancy of data from historical time series, the times associated with the total distance required to run through the entire monitoring network, and the sum of the times for each monitoring station. The method was applied to a hypothetical case-study consisting of a monitoring network with 32 stations. Cost-benefit analysis was performed to determine the number of stations to include in the new design versus loss of information. This optimization problem was solved with simulated annealing. Results showed that the relative reduction in exploitation costs more than compensates for the relative loss in data representativeness.

4.2 INTRODUCTION

The problem of reducing the dimension of an existing groundwater monitoring network (GMN) is addressed in this section. Both the quantity and quality of the data and the exploitation costs are included in the objective function. Data quality includes the representativeness of the parameter spatial field and temporal variation. The parameter spatial field is considered to be representative for the phenomenon under study if, with the amount of information available, the spatial variation is *best* reproduced with a well-defined classification method. Three methods have been proposed to obtain optimal GMN networks: i) variance reduction; ii) coupled simulation-statistical analysis; iii) information transmission (transinformation). The variance reduction (VR) methods were introduced in early works by Delhomme (1978), who used the so-called fictitious point method (generally used to assess the quality of covariance models when estimating with kriging), to determine the optimal location of rain gauges.

VR methods use the variance of the estimation error (σ_E^2) as an indicator of the accuracy of the estimated values. In geostatistics, the mathematical definition of σ_E^2 means that its value does not depend on the actual values of the measured variables, but rather on the relative spatial distribution of the measuring locations. Therefore, one may use σ_E^2 to indicate the spatial distribution optimality of a sampling network by testing all the combinations of available sampling locations, and choosing the combination that minimizes σ_E^2. Some additional assumptions must be made about the probability-density function for the estimation error: the estimation errors at any location in space are normally distributed - this may not always be the case (Journel, 1987) - with zero mean and a standard deviation equal to the square root of the estimation variance. A VR method is applied in this section. A short review of the other two other optimization methods is given below.

The coupled simulation-statistical analysis includes a wide range of techniques that have arisen from the need to optimize the spatial distribution of containment or pump-and-treatment wells in groundwater pollution problems. Meyer and Downey (1988) proposed a method for determining the best location for monitoring wells, following the works of Massmann and Freeze (1987a;b) in a risk-cost-benefit analysis for waste management facilities. The method's intention was to select the networks that maximize the probability of detection in the face of uncertainty. However, its practical applicability is hindered by the extreme simplicity of the analytical model used by Meyer and Downey (1988). Since then the containment problem has been thoroughly studied (*e.g.*, Wagner and Gorelick (1989, Lee and Kitadinis (1991, Shafike *et al.* (1992), Woldt and Bogardi (1992), Tiedman and Gorelick (1993)). A review of optimization and decision analysis for aquifer restoration and contaminant migration-control through pump-and-treat was made in Freeze and Gorelick (1999). These methods also have high potential in the design of regional or local networks for reference level monitoring.

Information transmission methods are based on the entropy as defined by Shannon (1948), on marginal entropy (Shannon and Weaver, 1949), and on the definition of transinformation (Amorocho and Espildora, 1973). Entropy is a measure of uncertainty; marginal entropy is a measure of the uncertainty of one outcome given the knowledge of a second outcome related to the first; transinformation is the reduction of the original uncertainty of one outcome given the knowledge of the second. These methods were applied to water resources with good results (Amorocho and Espildora, 1973; Caselton and Husain, 1980; Harmancioglu and Yevjevich, 1987; Harmancioglu and Alspaslan, 1992).

The inclusion of temporal information in the design of monitoring networks has been addressed in the geostatistical context as part of the covariance/variogram model by Angulo and Tang (1999), Lebel *et al.* (1987), Buxton and Pate (1994), and Pardo-Igúzquiza (1998),

among others. The resulting models are however, both very complex and problem-specific: the climatological variogram, in the form of a linear function of a time-dependent, space invariant, scaling parameter and a variogram model, in the case of the first two and last references; or a variogram model consisting of a three-dimensional isotropic spherical structure, a one-dimensional zonal linear structure, and a one-dimensional cosine structure, in the case of Buxton and Pate (1994). The sampling strategy may be designed not only to improve the precision of the estimated field, but also to target the areas that exhibit critical estimated values (Rouhani and Hall, 1988; ASCE, 1990b), higher temporal variability, or special temporal features that distinguish them from the other spatial locations. This limitation in the former models can be overcome by explicitly incorporating a temporal component in the objective function (OF), as is proposed here.

Monitoring network designs is limited by budgetary constraints (except for very special cases), which adds important limitations to the dimension of a particular design with regard to the number of stations to be included. In groundwater monitoring networks the costs are investment and exploitation costs. If the design is made from scratch then the number, location, frequency of monitoring and characteristics of the wells can all be included in the design, and the problem is multiobjective: maximizing the information gained, while keeping the investment and exploitation costs at a minimum. However, if the objective is to reduce the dimension of an existing monitoring network, then exploitation costs can be included in the OF, and the problem is formulated using location as the only variable. Some authors have addressed this problem in the context of variance reduction to optimize rainfall data collection network designs (Bras and Rodríguez-Iturbe, 1976), to assess the effect of altering the size and configuration of a groundwater monitoring network for groundwater levels (Candela *et al.*, 1988), and to establish an optimal network design to estimate the areal averages of rainfall events (Pardo-Igúzquiza, 1998).

When reducing the number of stations of a GMN the problem is combinatorial: which sub-set of stations, ω, out of the original set, Ω, should be retained? If the number of stations is large then the dimension of the combinatorial problem may be exhaustively intractable. Pardo-Igúzquiza (1998) recently solved this problem for rain gauges using simulated annealing. Rouhani and Fiering (1986) also used variance reduction techniques to determine the number and position of groundwater monitoring stations, and analysed the robustness and resilience of these methods (robustness was defined by Matalas and Fiering (1977) as "the insensitivity of a system design to errors, random or otherwise, in the estimates of those parameters affecting design choice"; resilience in the definition of Shannon (1948) is "the ability of the system to accommodate surprises and to survive under unanticipated perturbations"). Rouhani and Fiering (1986) found a significant degree of instability in the parameters of the covariance function (parameter space), though this instability had negligible effect on the action space. Similar results were obtained by Shannon (1948). Rouhani and Hall (1988) proposed the incorporation of risk, defined as the weighted sum of the expected value and the estimation variance, in order to correct the "blindness" of the estimation variance to extreme values. Longer reviews on this theme may be found in Loaiciga *et al.* (1992), Dixon and Chiswell (1996), and Harmancioglu *et al.* (1999).

In this section the problem of reducing a GMN dimension is treated in the context of variance reduction, coupled with a measure of temporal redundancy, such that important local temporal variabilities are detected, and exploitation costs in the form of the time taken to travel between stations. The term *data quality* is used when referring to both spatial accuracy and temporal redundancy, and *sampling time* when referring to the sum of monitoring and travel times. An objective function was built to incorporate these four factors.

The selection of the new GMN design was made by a cost-benefit analysis of the gain in data quality and sampling times. The OF was optimized for several ω values with simulated annealing (SA), and the optimal ω value obtained.

Two new developments in GMN optimization are introduced in this article: ii) a new definition of temporal redundancy; ii) the minimization of sampling times by incorporating them directly in the OF.

The article is organized in the following way: in the second section an objective function model is proposed; in the third section the simulated annealing algorithm proposed to solve the optimization problem is described, and its application is discussed; the fourth section presents a hypothetic case-study to test the method; the results of the application are presented and discussed in the fifth section; finally, in the sixth section the most relevant conclusions are drawn.

4.3 OBJECTIVE FUNCTION

The new GMN size and spatial distribution must assure that both spatial and temporal variabilities are correctly included in the new design, and that exploitation costs (sampling time) are minimized.

It is better to consider all the information collected, rather than using statistical transformations of the data. Common statistical variables are: maximums, minimums, percentiles, average values, variance, or more evolved transformations like projection in factorial spaces and similarity (or correlation) coefficients. Most of these statistical

transformations have the advantage of reducing a large amount of data to a treatable number of parameters. Despite their particular advantages, these methods tend to enhance some particular features of the data, while disregarding others. One way to circumvent this *blindness* is to use raw data whenever possible. A mathematical time series function that makes no statistical transformation of the raw data is proposed and used in this article. The function is a measure of data redundancy and is used to obtain better spatial distributions of monitoring stations.

Spatial accuracy, temporal redundancy, monitoring times for each station and total travel time are incorporated in a single OF and subjected to minimization. Simulated annealing (SA) is proposed to optimize the OF.

4.3.1 Spatial accuracy

Monitoring network optimization for geo-variables, such as in areal mean rainfall events, was the subject of early works by Rodríguez-Iturbe and Mejía (1974a;b), in which they considered the spatial and temporal variability of mean rainfall. The variance of mean rainfall was calculated as the product of point process variance, a reduction factor given by sampling in time (dependent only on the correlation in time and length of the time series), and a reduction factor given by sampling in space (dependent on the spatial correlation structure, the sampling geometry and the number of stations). These authors studied random and stratified random sampling schemes, and obtained an abacus for different correlation functions, the number of stations, and area of the region. Lenton and Rodríguez-Iturbe (1974) also considered the density and location of the stations. Bras and Rodríguez-Iturbe (1975) continued the latter's work, comparing former results for different point variances, the covariance function and covariance function parameter. In the same formalism, Bras and Rodríguez-Iturbe (1976)

included the cost associated with each station to help choose the best set (number and position) of stations with least cost and least mean rainfall variance. The methods based on the reduction of the estimation errors variance are known as variance-reduction techniques.

At this stage the variance-reduction method is developed using geostatistical nomenclature. Within the geostatistical formalism, field data are held to be the result of random processes of regionalized variables, *i.e.*, of random variables with space coordinates, with some spatial covariance. Regionalized variables are continuous in space, and therefore not completely random, but at the same time it is not possible to model them by means of a deterministic function (or spatial process). They therefore result from deterministic and stochastic processes (Matheron, 1970), incorporating the notion of uncertainty in the conception of inference models or in the simulation of variables (Matheron, 1970; David, 1977; Journel and Huijbregts, 1978).

The values of $z(x)$ at the sampled points in the field can be considered as realizations of a set of random variables $Z(x)$ in a field Γ. A set of random variables $Z(x_i)$ defined in a field Γ is a random function $Z(\boldsymbol{x})$:

$$Z(\boldsymbol{x}) = \{Z(\boldsymbol{x}_i)\}_{\boldsymbol{x}_i \in \Gamma}$$

and $z(\boldsymbol{x}_i)$ is a realization of the random variable; the latter a realization of a random function.

As field data are usually scarce, it is not possible to calculate the distribution function, but for practical applications only the first two moments are needed. Some restrictions with respect to stationarity are needed. The most common theory considers that the distribution function is invariant by translation. As, strictly speaking, the restrictive hypotheses are applied only to the first two moments, it is required that these exist and that they are independent of space coordinates (second order stationarity), otherwise the spatial covariance of the variable Z depends solely on the separation vector $\boldsymbol{h}$ (in modulus and direction). In this case only the

spatial increments have to be stationary (intrinsic stationarity). If these increments are made at step h, then the resulting expression is called the variogram:

$$\gamma(\boldsymbol{h}) = \frac{1}{2} E\left\{\left[Z(\boldsymbol{x}+\boldsymbol{h}) - Z(\boldsymbol{x})\right]^2\right\} \qquad (4.1)$$

The matrix system used to compute the kriging weights has to be invertible. For this property to be true, all the covariances of the system must be defined from positive definite functions (see, *e.g.*, Matheron (1970), pg. 54). The variogram has tended to replace covariance, mainly because of the less restrictive intrinsic hypothesis. In most practical cases this less demanding restriction on the random function model will not change the results (Deutsch and Journel, 1992).

To estimate the average value of a variable in an area A from values at locations x_α, $Z(x_\alpha)$, inside or outside the area, this value is equal to

$$V = \frac{1}{A}\int_A Z(x)\,dx \qquad (4.2)$$

A linear estimation of V can be obtained from n data points by

$$\widehat{V} = \sum_{i=1}^{n} \kappa_i \cdot Z(x_i) \qquad (4.3)$$

which is unbiased if the sum of the weights κ is one. This is a common requirement in several methods and also in kriging. This method is chosen because values of κ are determined so as to minimize the variance of the error of estimation. The kriging system is defined by (Journel and Huijbregts, 1978):

$$\begin{cases} \sum_{k=1}^{n} \kappa_k \cdot \gamma(h_{ij}) + \mu = \gamma(h_{iA}) \quad i = 1,\ldots,n \\ \sum_{k=1}^{n} \kappa_k = 1 \end{cases} \tag{4.4}$$

where μ is the Lagrange parameter, and $\gamma(h_{iA})$ is the average variogram between the point i and the area A when one extreme of the vector h is fixed in x_i and the other extreme describes the area A independently. The average variogram is then:

$$\gamma(h_{iA}) = \frac{1}{A} \int_A \gamma(x_i, u)\, du \tag{4.5}$$

Approximated numerically by

$$\gamma(h_{iA}) \approx \frac{1}{M} \sum_{k=1}^{M} \gamma(x_i, x_j) \quad x_j \in A \tag{4.6}$$

with M number of points used to discretize A.

The estimation variance is expressed by (Journel and Huijbregts, 1978):

$$\sigma_E^2 = \sum_{i=1}^{N} \kappa_i \cdot \gamma(h_{iA}) - \gamma(h_{AA}) + \mu \tag{4.7}$$

and

$$\gamma(h_{AA}) \approx \frac{1}{M^2} \sum_{i=1}^{M} \sum_{k=1}^{M} \gamma(x_i, x_j) \;\; x_i, x_j \in A \tag{4.8}$$

The estimation variance is a measure of the estimation accuracy of V. Because σ_E^2 only depends on the data points' location, and once a variogram model is defined, it is possible to change data location and calculate the estimation variance again. The spatial arrangement of points that minimizes σ_E^2 has the lowest estimation error, and therefore best reflects the spatial correlation introduced into the variogram model.

4.3.2 Temporal redundancy

All time series are considered synchronous and complete, either because data were collected at the same time, or because the necessary interpolations were made to synchronize data and fill in the gaps. Time events can therefore be handled as realizations of random functions, $Y_i(m)$, $i=1,\ldots, L_{FIXED} + L_{EXP}; m= 1,\ldots,D,$ (with L_{FIXED} being the number of fixed stations, *i.e.*, those that are to be included in all solutions; L_{EXP} the dimension of the subset of stations to be included in the new design; D length of the time series vectors). $Yi(m) \in X \in \chi$, with X being the current solution, and χ the solution space.

If the time series are not synchronous and/or complete then it is proposed that: i) data should be interpolated, or ii) a measure of temporal correlation be used, in either the frequency or correlation domains.

The sum of differences between time series, S, is used to evaluate redundancy: large values of S indicate that the series are very different, while smaller values indicate the opposite. S is an approximation to the ω-integral of time series functions.

The series are first normalized by subtracting the average value, $\overline{Y_i}$: $Y_i^0(m) = Y_i(m) - \overline{Y_i}$. Then the summation of the difference between the series is made for all times, with the possibility of shifting one time series in relation to the others by the time value d so that the sum is the lowest, *i.e.*, when summing the difference between series $Y_i^0(m)$ with series $Y_k^0(m)$ $k \neq i$, only the minimum values are used. This time translation should guarantee that time series are compared in-phase. Minimum temporal redundancy implies maximum S:

$$S(n) = Max\left\{ \sum_{\substack{i=1 \\ k=i+1}}^{L_{FIXED}+L_{EXP}-1} Min\left\{ \frac{\sum_{m=1}^{D-d} Y_i^0(m) - Y_k^0(m+d)}{D-d}, 0 \le d \le +\Delta \right\} \right\} \quad (4.9)$$

where $+\Delta$ is the maximum allowed time shift.

4.3.3 Exploitation costs

The costs associated with maintaining a GMN are considered to be dependent only on the time needed to measure at each station, in the form of equal or different times for measuring per station, and the time needed to travel through the whole network along the shortest path. Other costs not included in this formulation are, *e.g.*, maintenance costs and the cost associated with the non-detection of a pollution event (as considered, *e.g.*, in Massmann and Freeze (1987a;b), Wagner and Gorelick (1987), Rouhani and Hall (1988), Meyer *et al.* (1994)). The sum, UC, of individual measuring times, uc_i, i = 1,…, ω, is used to calculate the OF model,

$$UC = \sum_{i=1}^{\pi} uc_i \quad (4.10)$$

The determination of minimum total travel time (TC) is a well-known combinatorial problem known as the traveling salesman problem (TSP), solved here with the algorithm proposed by Carpaneto *et al.* (1995b). The optimal TC obtained by the algorithm is used to calculate the OF.

4.3.4 Optimization model

The network of ω stations with higher spatial accuracy, lower temporal redundancy, and lower exploitation costs (times), implies the minimization of the following objective function,

$$\begin{aligned} &\mathrm{OF} = w_1 \cdot \frac{\sigma_E^2}{Max(\sigma_E^2)} + w_2 \cdot \left[- \frac{\mathrm{S}}{Max(\mathrm{S})} \right] + w_3 \cdot \frac{\mathrm{UC}}{Max(\mathrm{UC})} + w_4 \cdot \frac{\mathrm{TC}}{Max(\mathrm{TC})} \\ &Conditional\, to: \\ &\mathrm{UC} + \mathrm{TC} \le 7\, hours \\ &available\, data \end{aligned} \qquad (4.11)$$

Max(σ_E^2), Max(S), Max(UC), and Max(TC) are the maximum values found during the optimization process, and are needed to make the OF dimensionless and problem-independent. It is assumed that the monitoring takes only one working-day of seven hours. This can be easily altered if necessary, however. The weights w_j, j = 1,…,4, are such that $\Sigma\, w_j = 1$. This may be useful if one wishes to give more emphasis to the quality of the data collected, by giving higher weights to the first two variables, or to control the exploitation costs more stringently, by giving higher weights to the two remaining variables. In the example given here, equal weights (=1) are used for all variables. We recall the comment made in section 3.3.3. about the decision to solve this problem as a single objective one.

4.4 SOLVING THE MODEL

Reducing a monitoring network of dimension Ω to a network of smaller dimension, ω, is a problem for which the number of possible combinations is given by $\Omega!/(\omega! \cdot (\Omega - \omega)!)$. However, if the dimension of the original network is large, there may likewise be a very large number of combinations, and computing the OF exhaustively would require much more time than is usually available. Therefore a more efficient method has to be used in order to obtain a good quality solution (in the sense that it may not be the global optimum) in a reasonable amount of time.

It is proposed to solve the problem by replacing one station at a time, evaluating the result, keeping the station if it reduces the OF or if the result fulfils a probabilistic criterion (the Metropolis criterion), and rejecting the station otherwise. The iterative process of replacing the stations and analyzing the fulfillment of the Metropolis criterion is a crucial part of the simulated annealing algorithm. The algorithm looks at each iteration for the cost of a given set of stations (a feasible solution), and it may be that a station that is rejected in one iteration, with a particular combination of stations, is accepted in a later stage with a different combination (different solution). Therefore an efficient search of the solution space has to be ensured. A global optimal solution would be found if the entire space was searched. However, in practical applications, if the number of combinations is very large, a global search would take far too much time. Algorithms that do not guarantee a global search can only provide optimal local solutions. This is the case with the simulated annealing (SA) implementation used here.

SA is one of the threshold algorithms included in the class of local search algorithms. The other two, as defined by Aarts and Korst (1990), are: iterative improvement, where only OF-reducing neighbors are accepted; and threshold accepting, where a certain deterministic non-increasing threshold sequence is used, allowing neighbor solutions with larger OF to be accepted, though to a limited extent. This is because the threshold value is fixed and always decreasing, with a very rigid control on the size of the cost difference. Simulated annealing uses a more flexible control on the threshold values, allowing transitions out of a local minimum at nonzero temperatures.

SA was first introduced by Kirkpatrick *et al.* (1983) as an algorithm to solve very well-known combinatorial optimization problems, reducing the risk of falling prematurely into local minima (or metastable solutions) common to iterative improvement methods, because they tend to accept only solutions that lower the OF, with fast quenching from high temperatures to

temperatures near zero. These authors proposed the use of the Metropolis (Metropolis *et al.*, 1953) procedure from statistical mechanics. This procedure generalizes iterative improvement by incorporating controlled uphill steps (to worse solutions). The procedure states the following: consider one small random change in the system at a certain temperature; the change in the objective function is ΔOF; if $\Delta OF \leq 0$, then the change in the system is accepted and the new configuration is used as the starting point for the next step; if $\Delta OF > 0$ then the probability that the change is accepted is determined by $P(\Delta OF) = exp(-\Delta OF / k_b T)$; a random number uniformly distributed in the interval (0,1) is taken and compared with former probability; if this number is lower than $P(\Delta OF)$ then the change is accepted. The control parameters $k_b T$ are replaced by the parameter t (also called temperature), to avoid using the Boltzmann constant, k_b, which would have no meaning in this context.

The SA algorithm runs in the following way: i) the system is *melted* at a high temperature (initial temperature, t_0); ii) the temperature is decreased gradually until the system *freezes* (no further OF change occurs); iii) at each iteration the Metropolis procedure is applied.

The generic SA algorithm for minimization, considering a neighborhood structure N, a solution space χ, and an objective function OF has the following pseudo-code.

```
Select an initial solution X_best;
Select an initial temperature t_0>0;
Select a temperature reduction factor;
Repeat
  Repeat
    Randomly select X ∈ N(X_best);
    δ = OF(X) – OF(X_best);
      if δ < 0 then
          X_best = X
      else
        generate random z uniformly in (0,1);
        if z < exp(-δ / t) then X_best = X;
  Until iterations = max_iterations
  Set t = α(t);
Until stopping condition = true;
X_best is the optimal solution found.
```

Once a feasible solution is generated it is passed to the OF calculation module (Figure 4.1). Here, two routines are internal (Redundancy and Monitoring time) and the other two are external, and were developed elsewhere. The following data is read: i) (*x,y*) coordinates and data values matrix into the kriging routine; ii) time series values matrix into the redundancy routine; iii) monitoring times per station matrix into the monitoring time routine; iii) travel distance matrix into the TSP routine.

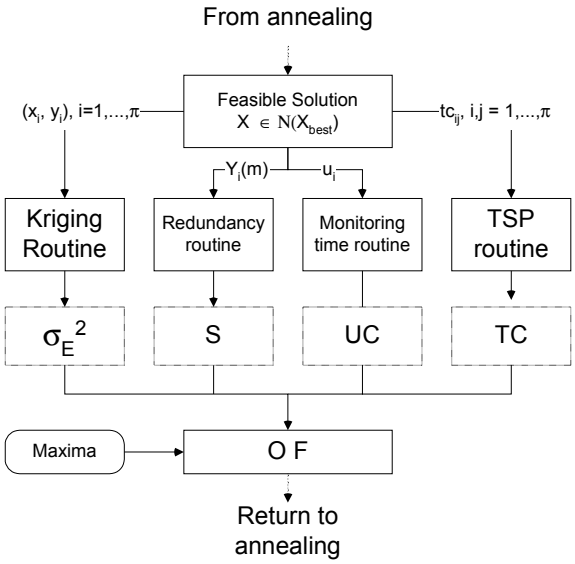


Figure 4.1. Objective function computing module

In order to speed-up the process several improvements have been proposed, particularly, limiting the number of iterations at each temperature, *i.e.*, defining the *max_iterations* number. The dimension of the Markov chain has been proposed as a function of the dimension of the problem (Kirkpatrick *et al.*, 1983): temperature is maintained until 100Ω solutions (iterations), or 10Ω successful solutions have been tested, whichever comes first. Ω stands for the number of variables (stations) in a problem. These authors also proposed that the annealing is stopped (stopping criterion) if after three consecutive temperatures the number of acceptances is not achieved. Alternatively, if the average value of the OF does not change after

a pre-established number of temperature decreases (R_{STOP}), then the annealing is stopped. Along with these dynamic criteria, a static one may be used to halt the process when a minimum temperature, t_{min}, is reached. This last criterion will guarantee that the annealing stops if none of the dynamic criteria is fulfilled, even before the total number of iterations is attained. In the present algorithm, both the dynamic and the static criteria were implemented.

Cunha and Sousa (1999) proposed the following expression to calculate the initial temperature, t_0,

$$t_0 = -\frac{b \cdot \mathrm{OF}_0}{\ln a} \qquad (4.12)$$

where OF_0 is the cost of the initial configuration, a is the elasticity of acceptance, and b is the amount of dispersion around the cost of the initial solution. Prior runs of the model determine the last parameter. The elasticity of acceptance represents the probability of accepting a solution worse than the initial one. The initial temperature determined by equation (4.12) is such that there is a probability a of accepting solutions that are b% worse than the initial solution. Temperature is usually decreased at a constant rate (cooling factor), α, such that after s temperature decreases the temperature is $t_s=t_0\alpha^s$. The two stopping criteria: t_{min} and number of temperature decreases are complementary because it is easy to calculate the minimum temperature attained if t_0 and α are known.

The generation mechanism of a new solution, U, from the neighborhood, $N(X)$ of a solution X is a simple one: Consider the solution vector $X=\{x_1, \ldots, x_\omega\}$; i) randomly remove one element (station) from X; ii) randomly choose an element in $\chi \setminus X$; iii) insert this element in X, obtaining U; iv) confirm if U is a valid candidate solution (eq.4.11), if yes continue, otherwise goto i).

A specific computer code in FORTRAN was developed by the authors to solve GMN reduction problems (MINCOST). The code incorporates two routines developed by others, namely the

kriging routine (Deutsch and Journel, 1992), and the solution for the traveling salesman problem (Carpaneto *et al.*, 1995a;b). The code was thoroughly tested and validated.

4.5 CASE STUDY

In order to test the method, a grid of 32 monitoring stations randomly distributed in a square grid of eight by eight spatial units was created (Figure 4.2). Monitoring stations numbers 1 and 2 represent wells for water supply and are therefore included in all the solutions. Ω is therefore equal to 30. Time series of generic geo-variables for all stations were calculated using common mathematical functions, purely deterministic and with normally distributed errors. The functions were selected based on empirical judgment and experience, attempting to simulate the behavior of water quality variables. Some of the resulting variables are non-homocedastic, thus reflecting a feature common to variables such as redox potential and electrical conductivity when the scale of measure has to be changed between consecutive observations. The unit of the generic geo-variable is identified by un. The time series equations are presented in Table 4.1, where τ stands for unitary measuring interval (*e.g.*, one day, one months), $k_1\tau$ stands for the first record moment and $k_n\tau$ for the last, k= 1, 2, ….

Table 4.1. Time series equations

Series number	Equation: x =
1	$Cos(k\tau)+0.54\ k\tau$
2	$0.12\ k\tau$
3	$Sin\ (k\tau)$
4	$Cos((k+1)\ \tau)$
5	$Cos((2k+1)\ \tau)$
6	$Log(k\tau) – Log(k_n-k_1)$
7	$Exp[-k\tau\ /\ (k_n-k_1)]$
8	1
9	$Cos(0.3\ k\tau + 3\ (k_n-k_1))$
10	$Atan(k\tau)$
11	$Atan(k\tau + 1.5\ (k_n-k_1))$
12-22	Repeat 1-11 + 20% normally distributed error around the mean
23-32	Repeat 1-10 + 40% normally distributed error around the mean

The intention was then to choose the ω monitoring stations that should be added to stations 1 and 2 in order to obtain a network that has optimal exploitation costs, best preserves the temporal features of the data, and has optimal spatial accuracy.

The first goal was to determine the number of stations to be included in the new design; the second goal was to identify the stations in the new optimal design.

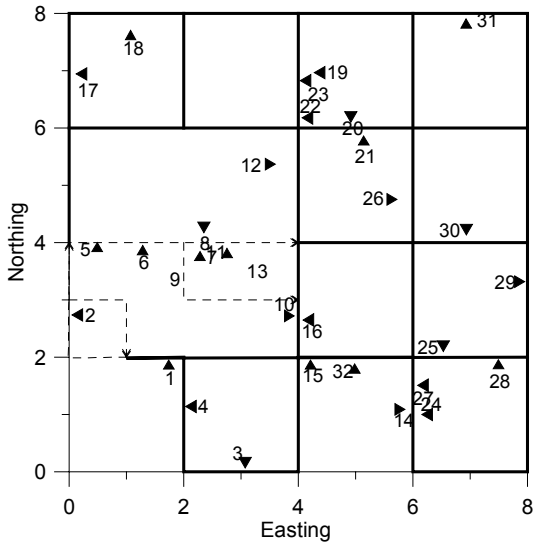


Figure 4.2. Location of the monitoring stations and routes connecting them: bold line for two-way streets and discontinuous line for one-way streets

4.6 RESULTS

4.6.1 Cost-benefit analysis

Eight different subset cardinalities of ω+2 stations were tested, {13,15,17,19,21,23,25,27}, and a cost-benefit analysis was performed for the gain in accuracy and temporal representativeness and the increase in exploitation costs (time needed to sample). The

estimation variance was calculated using an experimental spherical isotropic variogram for the first time period with the following parameters: nugget = 8.1×10^{-3} un^2; sill = 8.01×10^{-2} un^2; range = 2.482 m (a more detailed analysis is presented in section 3.8)

As expected, the estimation variance decreased as new stations were added (Table 4.2) (indicating an increase in spatial accuracy), but the gain in spatial accuracy was much more significant up to the 23rd station, with each new station producing an average increase in spatial accuracy of 8.1%; after the 23rd station the spatial accuracy only increased 0.5% each time a new station was added.

It should be noticed here that the interpretation of data redundancy (S) is different when considered in the context of cost-benefit analysis or in the context of network optimization for a pre-defined ω number. In the first case $S(\Omega+2)$ represents the total amount of information in the time series, and $S(\omega_2) - S(\omega_1)$, with $\omega_1 < \omega_2$, is the marginal S, or the amount of information added by the $\omega_2 - \omega_1$ stations. In the second case no marginal S is calculated, and S is always the total S, calculated for different $\omega+2$ values.

The gain in temporal representativeness (marginal S) increased every time a new station was added, but the average rate of increase was almost four times higher up to the 19th station than it was for the remaining stations (9.8% against 2.5%). Figure 4.3a shows the accuracy and representativeness curves with the fitted trend lines drawn as interrupted straight lines.

Both monitoring and travel times (UC and TC) increased polynomially (2nd degree) with the number of stations, with UC having a faster growth rate than TC. One interesting feature is the stabilization of travel times after the 23rd station; a similar stabilization is found for the monitoring time (UC) after the 21st station, but in this case UC increases again after the 23rd. As a result the increase in total sampling time shows a polynomial behavior up to the 21st

station, after which it becomes linear. Figure 4.3b shows the variation of UC, TC and UC + TC, with the number of stations and the adjusted second-degree polynomials.

These results show an interesting data quality/sampling time structure: up to the 23rd station, though data quality increases linearly, sampling times increase polynomially; after the 23rd station the increase in data quality is much lower and the sampling times still increase, but at a lower rate. This indicates that a monitoring network with 23 stations would be the optimal solution.

Sampling times cost structure may be investigated by analyzing the relative contribution of UC and TC to the total time. Figure 4.4 shows the relative contribution of UC to the total sampling time. UC is always the most important cost factor, and its importance increases with the dimension of the network, at least up to the 21st station, after which the relative weight tends to stabilize around 70%. The jump at the 23rd station is justified by the need to introduce new stations in locations with difficult access, for which the new path is longer; the subsequent increase indicates that the stations added after the 23rd can be connected by the new-found path, and UC becomes much more important again. Therefore, new gains in efficiency (reduction of total sampling time) should be sought by reducing the measuring times, u_i.

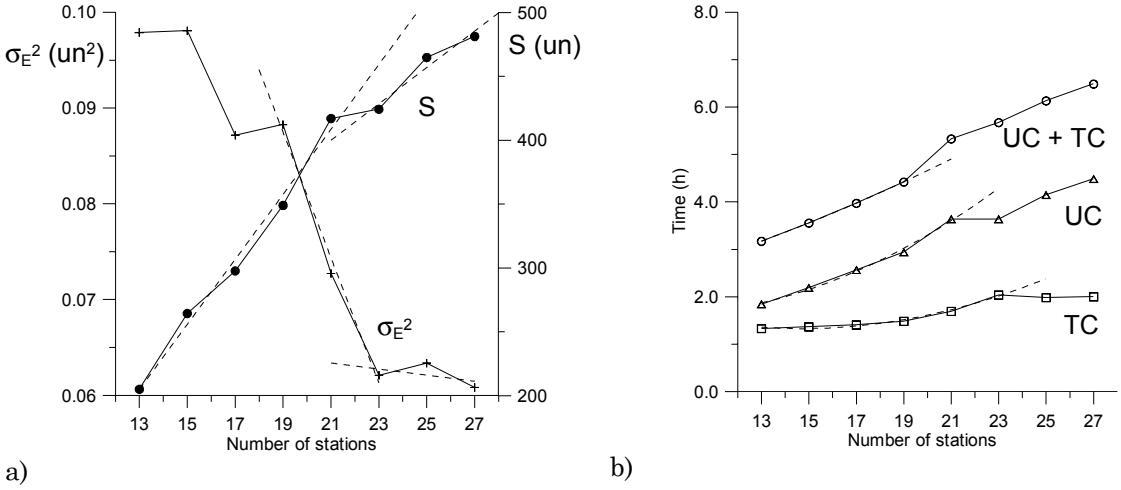


Figure 4.3. Cost-benefit analysis for increasing number of stations: a) gain in spatial accuracy and temporal representativeness; b) Time necessary to sample

Table 4.2. Optimization results for the eight ω values. The original GMN results are also shown

Number of Stations	OF	σ_E^2 (un²)	S (un)	UC (h)	TC (h)	UC + TC
13	1.885	0.09788	204.9	1.841	1.327	3.168
15	1.966	0.09808	264.3	2.190	1.362	3.552
17	1.915	0.08715	297.6	2.563	1.406	3.969
19	2.032	0.08827	349.0	2.935	1.482	4.418
21	2.053	0.07273	417.0	3.635	1.692	5.326
23	2.044	0.06209	424.3	3.634	2.036	5.670
25	2.021	0.06336	464.8	4.147	1.982	6.129
27	2.119	0.06081	481.3	4.485	1.999	6.483
32	-	0.05937	550.4	5.990	2.704	8.694

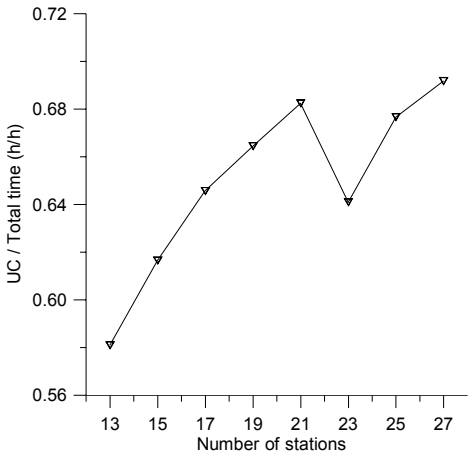


Figure 4.4. Relative contribution of UC to the total sampling time.

The selection of stations show high congruency for the different ω+2 values (see Figure 4.5): i) four of the stations are not included in any design (7, 9, 10 and 13); ii) four of the stations are only included for very high ω+2 values (≥ 25) (5, 6, 11, 25). The interpretation is complex due to the presence of four variability factors, but it is still possible to observe the following: the stations mentioned make a small contribution to the increase in spatial accuracy, but, on the other hand, they tend to increase the total travel time (their location being on one-way roads). Furthermore, the stations involve low marginal Ss (see Figure 4.3a and b)

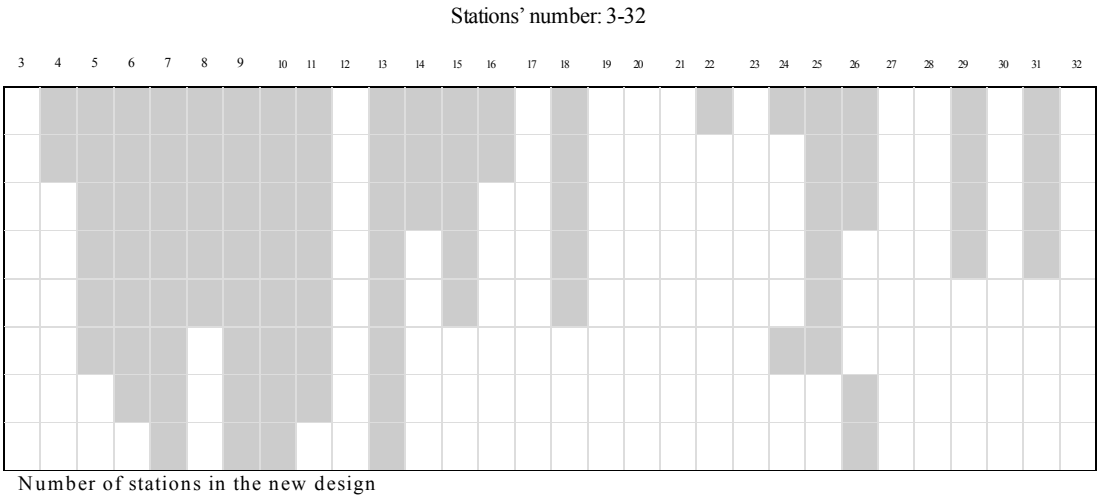


Figure 4.5. Stations excluded in the new designs (in gray)

4.6.2 Selected monitoring network

The selected ω+2 station GMN is now analyzed, but equivalent analyses could be made for different ω values, not included due to space limitations. Figure 4.6 shows the OF, data quality and sampling times convergence curves. The OF starts in the first iterations at high temperatures, with values close to 2.7, indicating that it is still far from its minimum. This is clearly visible in Figure 4.6, as the four OF variables slowly converge towards their optimal values. At sufficiently low temperatures a *frozen* system is achieved and the optimal (local) solution is reached. The solution GMN is shown in Figure 4.7.

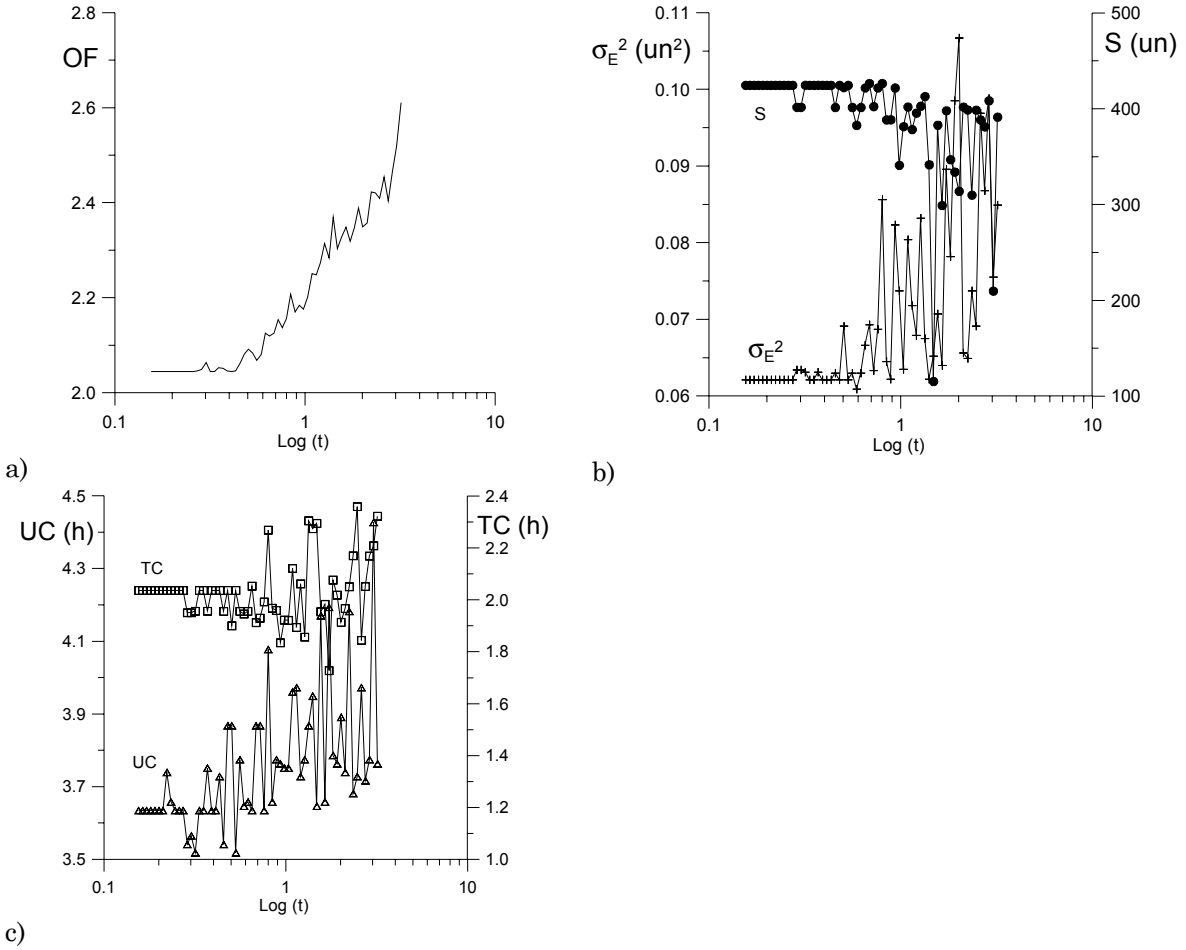


Figure 4.6. Optimization results for a monitoring network of 23 stations. a): Objective function; b) data quality variables; c) sampling times

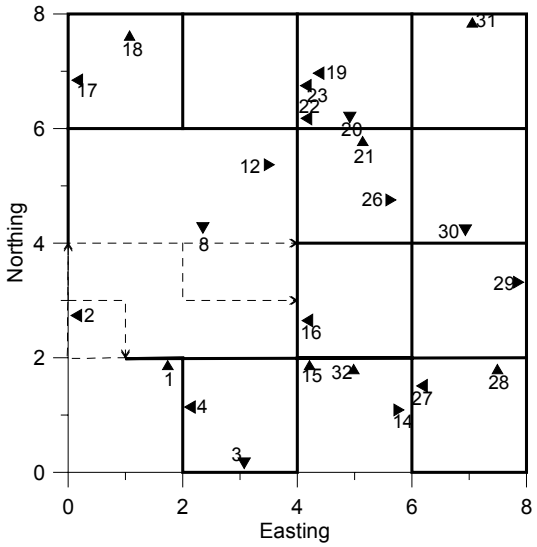


Figure 4.7. Resulting monitoring network with 23 stations

The resulting $\omega+2$ = 23 stations GMN is 28 % smaller than the original one, and results in a reduction of 3.0 hours in sampling time (35 %). At the same time, only 20% of spatial accuracy (0.0119 un^2) and 22.9% of total time series information (126.1 un) was sacrificed.

4.7 CONCLUSIONS

GMN optimization problems are particularly difficult to solve because of discontinuous integer-related nonconvexity (as a monitoring station is either included in the new design or not, the feasible region is discontinuous) and continuous nonconvexity of the interpolation error variance surface (due to the presence of many local minima). Some solvers can handle one or the other type of nonconvexity, but the combination of both requires heuristic methods. Because the aim of this article was to test the objective function model, no comparison of algorithms was made, rather, simulated annealing was selected for its good results in other optimization problems. Furthermore, it has already been tested for monitoring networks design against other heuristic algorithms, and it has outperformed the them (tabu search, genetic algorithms, and sequential exchange search algorithms) (Lee and Ellis, 1996).

In many instances it is neither possible nor advisable to incorporate statistics-based frequency methods in the optimization models due to the short length of time-series, missing data, and the bias introduced in the data by the empirical selection of locations and frequencies of the existing monitoring networks. An automatic frequency/location optimization will, in these cases, seriously compromise the quality of the solution. A better alternative is to establish the frequency beforehand, after judicious analysis of the available time series and the hydraulic properties of the aquifer (*e.g.*, in karst aquifers frequency must, in most cases, be higher than in porous aquifers). The location and number of stations are then selected afterwards. This is

the case of the GMN optimization problem solution described here for reference level/ compliance monitoring with a pre-set sampling frequency.

The method proposed in this article seems to have a good applicability for reference level ground water monitoring networks when the main spatial variability and behavior of time series should be preserved, but exploitation costs are to be optimized as well. Due to the definition of temporal redundancy, which is calculated in the data space rather than covariance space (Amorocho and Espildora, 1973; Caselton and Husain, 1980; Harmancioglu and Yevjevich, 1987; Harmancioglu and Alspaslan, 1992), there is a tendency to overburden time series with higher values and/or higher variability. This feature is, however, welcome in most situations, and is an accepted penalty for not having to consider particular variable or multivariate statistical distributions.

The results showed that it is possible to eliminate some stations from a GMN and have a relative lowering of costs greater than the relative reduction of GMN dimension and collected data quality.

5 GROUNDWATER NITRATE MONITORING NETWORK OPTIMIZATION WITH MISSING DATA

5.1 SYLLABUS

A method to design groundwater monitoring networks to detect agricultural contamination is proposed. The method is particularly well adapted to the reduction of existing networks where data is missing from time series records. Simulated annealing optimization algorithm is used to minimize the variance of the estimation error obtained by kriging, in combinatorial problems created by selecting an optimal sub-set of stations from the original set. Optimization is performed for several measuring times, obtaining an equal number of optimized small dimension networks; stations that repeat more often in these networks are selected to make part of the final network. A compliance groundwater nitrate monitoring network in the south of Portugal is used to illustrate the method. The original 89 stations network was converted into 16 stations. Results show that very important reductions in

exploitation costs (about 80%) are compatible with a cost-effective network capable of detecting non-compliance with national and European norms.

5.2 INTRODUCTION

This article focus on the design of a groundwater monitoring network (GMN) to detect contamination with nitrates from agricultural origin. This is an issue that has been in the mind of general public, scientists, governmental agencies and legislators for some time now. The concern is well justified if one looks at European statistics: with the still incomplete coverage of all the territory, more than 25% of the monitored regions pass the 25 mg/l national guide level for groundwater nitrate concentration (Portuguese Law No. 236/98) in all monitored stations, and in 13% of the regions the 50 mg/l European water quality standard for drinking water (Drinking Water Directive, 98/83/EC (EU, 1998)) is passed in more than 25% of the monitoring stations (Scheidleder *et al.*, 1999).

European Union produced legislation for groundwater protection in 1980 (Groundwater Directive, 80/68/EEC (EU, 1980)), and in 1991 a specific legislative framework for nitrate with origin in agriculture (Nitrate Directive, 91/676/EEC (EU, 1991)). The latter establishes goals for preventing pollution and decrease existing nitrate concentrations. To monitor the effectiveness of such actions Member States must include regular monitoring of groundwater quality. In Portugal a national groundwater monitoring network with hundreds of stations has been operating since the eighties, unfortunately with many gaps. Three major aquifer systems were considered particularly vulnerable to nitrate contamination and have been monitored in detail since 1996. The case-study presented in this article is in none of those systems, but shows, nevertheless, nitrate concentrations always above 50 mg/l. The existing monitoring network consists in 89 stations distributed randomly in a 50 km^2 area. The objective of the study is to reduce the number of stations to a more cost-effective network. A variance reduction based objective function is defined and optimized with an heuristic optimization algorithm (simulated annealing).

Given the high costs associated with the implementation of an environmental monitoring network, development of efficient procedures for its design is a fundamental task. Moreover network design depends upon the spatial and temporal distribution of physicochemical parameters, which may be unknown before sampling, or that may evolve into a different distribution due to any external cause (*e.g.*, introduction of some pollution source, or change of flow direction). Thus, a monitoring network should not be considered as a static measuring tool, rather, it should be checked for efficiency from time to time, and adjusted if needed. Loaiciga *et al.* (1992) divided the general types of approaches to network design into hydrogeologic, when no advanced geostatistical method is applied, and statistical otherwise. This classification, despite limiting because it excludes promising advanced statistical techniques such as those derived from information theory (Harmancioglu *et al.*, 1998), is useful and helps tracing a border line between techniques that do not incorporate spatial statistical information, and those that do. Loaiciga *et al.* (1992) also divided statistical methods into simulation, variance-based, and probability-based. The main difference between these methods lays in the formulation of the objective function to be optimized, and the difference between different approaches in the constraints and on the way optimization is undertaken. Probability-based methods include variations to the others by including the values taken by the state variable and the probability of missing detecting high values. In our review no distinction was made and probability-based methods were included in both simulation and variance-based methods.

Simulation methods consider the uncertainty in the hydraulic conductivity field and therefore uncertain head distribution and velocity fields. Different random hydraulic conductivity fields are usually estimated by conditional simulations given a spatial covariance model obtained with the existing field data. The resulting differences in the estimated velocity fields will influence the optimal distribution of monitoring stations according to some objective function

(*e.g.*, dependent on the probability of missing the detection of contamination). Examples of this method can be found in such works as Massmann and Freeze (1987b) and Wagner and Gorelick (1989).

Variance-based methods, also known as variance reduction methods, consider that the uncertainty associated with a given monitoring network may be determined by the variance of the estimation error obtained by kriging. A given spatial distribution of stations has an uncertainty that depends on the particular locations. If one station is removed or another is added the accuracy will usually decrease in the first case and increase on the second. Also, if the number of stations is kept and only their location altered, accuracy will change. Variance of the errors of estimation is therefore used as objective function. Here two alternatives have been proposed, one based directly on the kriging variance as calculated from the kriging system, and another based on the variance of the errors calculated with the estimated and known values at each point. Journel (1987) and Delhomme (1978) advocate that the first is not a measure of local uncertainty at the estimated points, and therefore should not be used with this objective; Delhomme (1978) proposes that the latter is used instead. Despite this controversy kriging variance has been extensively used for monitoring network design. Examples can be found in works by Bras and Rodríguez-Iturbe (1976), Rouhani (1985), Loaiciga (1989), Rouhani and Hall (1988), Pardo-Igúzquiza (1998). Rouhani and Fiering (1986) showed that these methods are robust ("insensitive to design errors, random or otherwise, in the estimates of those parameters affecting design choice") and resilient (as the "ability to accommodate surprises and to survive under unanticipated perturbations").

The reduction of the dimension of an existing monitoring network is an optimization combinatorial problem: which subset of stations should be kept in the new design. When the number of stations is large the number of possible combinations makes it impossible to test all possible combinations, even in very fast computers. One common practical decision is to accept one good solution even if it is not the optimum. Some heuristic optimization algorithms exist to

deal with combinatorial problems, namely, a general class of strictly descendent algorithms that include sequential exchange with node-swap or with node-substitution, downhill simplex, search with multiple randomly-generated starting solutions; and a general class of algorithms not strictly descendent that include simulated annealing and tabu search, or combinations of these. Other algorithms based on nature like genetic algorithms and ant colonies optimization (see, *e.g.*, an introductory article in Dorigo *et al.* (1996)), are showing very good results also. Most of these methods have been applied to monitoring network design with more or less success. Simulated annealing has however been more frequently applied to monitoring network design probably because it is the oldest of the non-strictly descendent methods and has shown to be very efficient (Lee and Ellis, 1996).

GMN optimization has been thoroughly applied essentially with two objectives: i) network augmentation and ii) original network design. Optimization for network reduction is a much less common application. Examples of the latter are the works by Knopman and Voss (1989), Meyer *et al.* (1994) and Reed *et al.* (2000), for point sources using flow and transport models. Grabow *et al.* (1993) propose a method for network reduction without the need to simulate mass transport and stated as general for both point and diffuse sources, though only applied by the authors to a point source. The method proposed in the present article is oriented to diffuse contamination, particularly to agricultural contaminants and applied to nitrogen (measured as nitrate).

5.3 CASE-STUDY

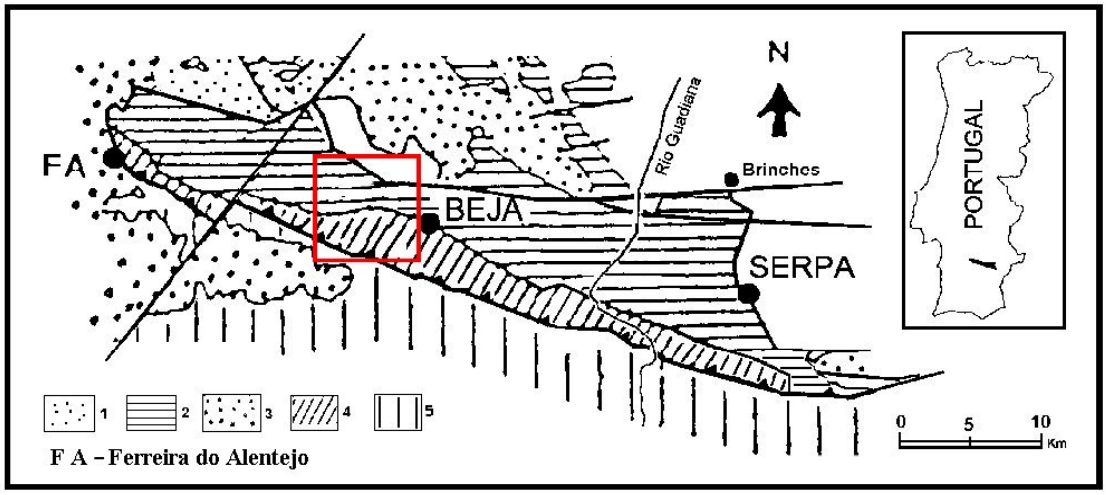
The case-study area is located in south of Portugal in an area known as Beja's Gabbro aquifer system developing between Ferreira do Alentejo (NW) and Serpa (SE) (Figure 5.1), covering an area of about 350 km^2 in Ossa-Morena geotectonic unit. Only the most relevant geologic and

hydrogeologic aspects will be reviewed here. Complete information on these and other themes are available in the thesis by Fonseca (1995), Duque (1997), and Peralta (2001).

The Gabbro-Dioritic shallow aquifer is one of the most productive formations of the Alentejo region when compared with other hard aquifers occurring in this region. The aquifer is unconfined; in the upper altered zones, down to about 30 meters, water circulation is through porosity, while after crossing unaltered rock, water circulation is mainly through secondary porosity.

In the study area wells productivities' range from 1 to 15 l/s with most frequent average values around 5 l/s. Most frequent transmissivity values are around 100 m^2/day. Recharge is estimated between 10% and 20% of average annual rainfall (584 mm/yr). Hydrochemical characterization indicates that these waters have high mineralization, with total dissolved solids ranging from 400 mg/l to 900 mg/l. The main water facies are calcium-magnesium bicarbonate and magnesium-calcium bicarbonate.

The actual groundwater quality is the long-term consequence of major changes in agriculture in Alentejo since 1930-1940, when cereal crops replaced autochthonous forest. Land is mainly occupied with wheat and sunflower or corn as alternative crop. Under intensive cereal cultivation the common fertilizer application is estimated between 100-120 kg N/ha/yr. Since de seventies high nitrate concentrations have been identified in some public wells in the vicinities of Beja. Average nitrate content in public and private wells is nowadays frequently above 50 mg/l.



Legend: 1-Odivelas Volcano-sedimentary Complex. 2-Maphic and intermediate plutonic rocks (Beja's Gabbro Complex). 3-Plutonic acid and sub-volcanic rocks (Baleizão Porphirs). 4-Beja-Acebuches Ofiolitic Complex (meta-gabbros and basic meta-volcanites). 5-Pulo do Lobo Accretion Terrain (South Portuguese Geotectonic Unit-schists). Geology adapted from Fonseca, 1995).

Figure 5.1. Geographical location of Beja's Gabbro aquifer (350 km^2) and study area (50 km^2).

5.4 METHODS

5.4.1 Reducing the dimension of an existing groundwater monitoring network

The problem of reducing an existing groundwater monitoring network is proposed to be solved by i) define a fixed number of stations to be included in a *first-approximation GMN*; ii) at each measuring time estimate the subset of stations from the original complete set that has the highest accuracy given the *first-approximation GMN*; iii) repeat the previous step for all measuring times; iv) compute the frequency with which stations appears in the estimated *first-approximation GMN* sets; and v) include the most frequent stations in the final GMN. See

First-approximation GMN dimension may be a tricky parameter because if there are too many missing data in the time series then a too low number of stations may lead to very small final GMN, and a too high number to very large final GMNs. A good approximation is to use a value

similar to the intended final GMN, which can be determined by classical statistical methods for sampling (Thompson, 1992; Cochran, 1977). Step ii) is proposed to be handled in the context of variance reduction techniques coupled with optimization by simulated annealing (SA). Estimation error variance is the measure of accuracy used as the objective function to be minimized by SA when iteratively selecting subsets from the original complete set. Both methods are discussed in the next sections in more detail. *First-approximation GMN* optimization is then made for each measuring time. In step iv) a final GMN is obtained by choosing the set of stations that appear most frequently in the optimized *first-approximation GMN*s,

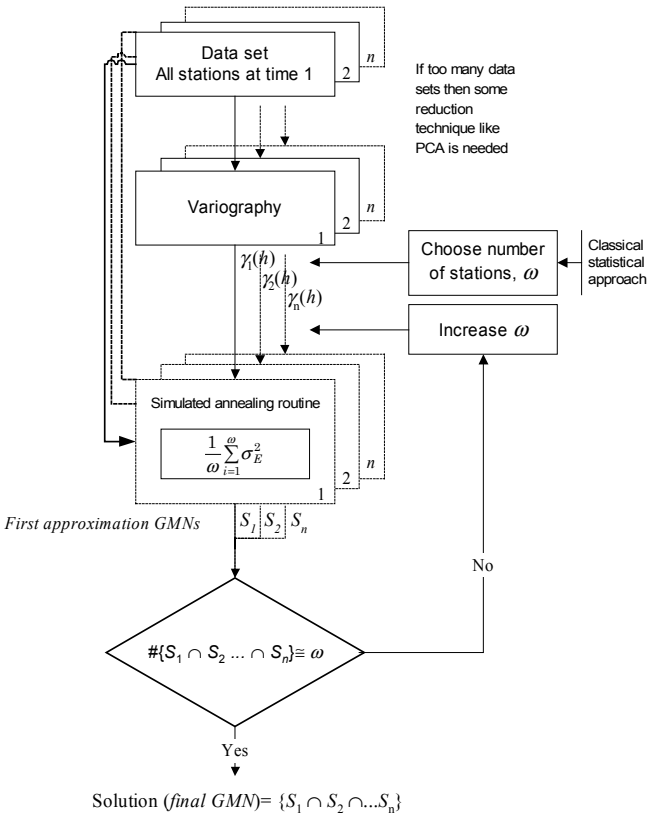


Figure 5.2. Proposed method

If time series data are complete, in the sense that a very small fraction of values is missing, other variance reduction methods for GMN optimization may be used instead, namely those based on the space-time analysis (Buxton and Pate, 1994; Pardo-Igúzquiza, 1998) and on time-only (Amorocho and Espildora, 1973; Caselton and Husain, 1980; Harmancioglu and Yevjevich, 1987).

5.4.2 Estimation error variance

The values of a state variable *z(x)* at the sampled points in the field can be considered as realizations of a set of random variables *Z(x)* in a field Γ. A set of random variables $Z(x_i)$ defined in a field Γ is a random function $Z(\mathbf{x})$:

$$Z(\mathbf{x}) = \{Z(\mathbf{x}_i)\}_{\mathbf{x}_i \in \Gamma}$$

$z(\mathbf{x}_i)$ is a realization of a random variable, and the latter a realization of a random function. The most common theory considers that the distribution function is invariant by translation. In rigor the restrictive hypotheses are applied only to the first two moments, hence it is only required that they exist and be independent of space coordinates (second order stationarity), or, the spatial covariance of the variable Z be dependent only on the separation h between two coordinates. In this case only the spatial increments have to be stationary (intrinsic stationarity). If these increments are made at step h, then the resulting expression is called the variogram.

An estimate of the mean value of a state variable in an area A is

$$V = \frac{1}{A}\int_A Z(x)\,dx \qquad (5.1)$$

A linear estimation of V can be obtained from p data points by

$$\hat{V} = \sum_{i=1}^{p} \kappa_i \cdot Z(x_i) \tag{5.2}$$

which is unbiased if the sum of the weights κ_i is one. This is a common requirement in different estimation methods, and also in kriging. This method is chosen because the κ are determined so as to minimize the variance of the error of estimation. The kriging system is (Journel and Huijbregts, 1978):

$$\begin{cases} \sum_{i=1}^{n} \kappa_i \cdot \gamma(h\ ) + \mu = \gamma(h_{iA}) \\ \sum_{i=1}^{n} \kappa_i = 1 \end{cases} \tag{5.3}$$

where n is the number of samples used to estimate the value at A, μ is the Lagrange parameter, and $\gamma(h_{iA})$ is the average variogram between the point i and the area A when one extreme of the vector h is fixed in x_i and the other extreme describes the area A independently. The average variogram is:

$$\gamma(h_{iA}) = \frac{1}{A} \int_A \gamma(x_i, u)\, du \tag{5.4}$$

which may be approximated numerically by

$$\gamma(h_{iA}) \approx \frac{1}{M} \sum_{j=1}^{M} \gamma(x_i, x_j) \quad x_j \in A, x_i \in \Gamma \tag{5.5}$$

with M the number of points used to discretize A.

The optimal estimation variance is expressed by (Journel and Huijbregts, 1978),

$$\sigma_E^2(A) = \sum_{i=1}^{n} \kappa_i \cdot \gamma(h_{iA}) - \gamma(h_{AA}) + \mu \tag{5.6}$$

The estimation variance is a measure of the estimation accuracy of V. Because $\sigma_E^2(A)$ only depends on the geometric configuration of the data points, and, once a variogram model is defined, it is possible to change data locations and calculate the estimation variance again.

The spatial arrangement of points that minimizes $\sigma_E^2(A)$ has the lowest estimation error, and therefore best reflects the spatial correlation introduced in the variogram model.

As the estimation variance is a local measure of accuracy (at *A*), some global measure is necessary to allow comparing two alternative spatial configurations. Woldt and Bogardi (1992) proposed the average measure of estimation variance weighted by prior suspicion of the value $\hat{V}$, and the average measure of kriged contamination level weighted by estimation variance as the global measure. Here an equal measure is used, but considering equal weights, *i.e.*, a simple mean of the estimation variance,

$$\sigma_E^2 = \frac{1}{\omega - 1} \sum_{i=1}^{\omega} \sigma_E^2(A_i) \qquad (5.7)$$

where A_i denote each one of the areas to be estimated, and ω is the number of areas to be estimated.

After setting the number of stations, ω, a method of leave-one-out is used to calculate the estimation variance at each station (location), by solving the kriging system ω times with ω-1 stations.

Equation (5.7) reflects the accuracy of estimating with a set of stations, and the most accurate the estimates, the lowest the mean estimation variance. Therefore, stations where the estimation variance is low will be preferred to others with higher.

5.4.3 Optimizing a monitoring network

The optimization problem can be stated in the following way: what is the "best" subset of ω stations (chosen according to their location), out of the initial set of Ω stations, in terms of the mean estimation variance?. Which can be stated as

$$\min \sigma_E^2 \tag{5.8}$$

If the number of stations is small then all combinations of ω in Ω can be tested; however if it is large the number of combinations becomes very large and the problem is classified as a difficult combinatorial optimization problem, for which an exhaustive search of all possible combinations is not possible in a reasonable amount of time. Solutions to these problems may however be approached by heuristic algorithms that iteratively look for better solutions by trial and error. One of such algorithms is the well-known simulated annealing (SA). It is one of the threshold algorithms included in the class of local search algorithms. The other two are, as defined by Aarts and Korst (1990): iterative improvement, where only σ_E^2-reducing neighbors are accepted, and threshold accepting, where some deterministic non-increasing threshold sequence is used, allowing neighbor solutions with larger σ_E^2 to be accepted, but in a limited way because the threshold value is fixed and always decreasing with a very rigid control on the size of the σ_E^2 difference, $\Delta\sigma_E^2$. Simulated annealing uses a more flexible control on the values of the threshold, allowing transitions out of a local minimum at nonzero temperatures.

SA was first introduced by Kirkpatrick *et al.* (1983) as an algorithm to solve very well known combinatorial optimization problems, reducing the risk of falling into local minima (or metastable solutions) common to iterative improvement methods. These authors proposed the use of the Metropolis *et al.* (1953) procedure from statistical mechanics. This procedure generalizes iterative improvement by incorporating controlled uphill steps (to worse solutions). The procedure states the following: consider that the change in the objective function is $\Delta\sigma_E^2$; if $\Delta\sigma_E^2 \leq 0$, then the change in the system is accepted and the new configuration is used as the starting point in the next step; if $\Delta\sigma_E^2 > 0$ then the probability that the change is accepted is determined by $P(\Delta\sigma_E^2) = exp(-\Delta\sigma_E^2 / t)$, where t is a control parameter called temperature; a random number uniformly distributed in the interval (0,1) is taken and compared with the

former probability; if this number is lower than $P(\Delta\sigma_E^2)$ then the change is accepted. The control parameters t is usually termed temperature. The SA algorithm runs in the following way: i) the system is *melted* at a high temperature (initial temperature, t_0); ii) the temperature is decreased gradually until the system *freezes* (because no better solutions are found and the probability of uphill steps is near zero); iii) at each iteration the Metropolis procedure is applied; iv) if any of the stopping criteria is reached the algorithm is stopped and the best solution found is presented. The generic SA algorithm for a minimization, considering a neighborhood structure N, and an objective function σ_E^2 has the following pseudo-code.

```
Select an initial solution X_best;
Select an initial temperature t0>0;
Select a temperature reduction factor;
Repeat
  Repeat
    Randomly select X ∈ N(X_best);
    Δσ²_E = σ²_E (X) – σ²_E (X_best);
      if Δσ²_E <0 then
                X_best = X
      else
                generate random z uniformly in (0,1);
      if z<exp(-Δσ²_E /t) then X_best = X;
  Until iterations = max_iterations
  Set t = α t;
Until stopping condition = true;
X_best is the optimal solution found.
```

In order to speed-up the process several improvements have been proposed, namely by limiting the number of iterations at each temperature, *i.e.*, defining the number *max_iterations*. The dimension of the Markov chain has been proposed to be a function of the dimension of the problem (Kirkpatrick *et al.*, 1983): temperature is maintained until 100Ω solutions (iterations), or 10Ω successful solutions have been tested, whichever comes first. Ω stands for the number of stations in the problem. These authors proposed also that the annealing is stopped (stopping criterion) if after three consecutive temperatures the number of acceptances is not achieved. It can also be considered that if the average value of the OF does not change after a pre-established number of temperature decreases (R_{STOP}), then the annealing is stopped. These

parameters control the time spent at each temperature and the total running time. Along with these dynamic criteria, a static one may be used to halt the process when a minimum temperature, t_{min}, is reached. The former will ensure that the annealing will stop if none of the dynamic criteria is fulfilled, even before the total number of iterations is attained. In our algorithm both the dynamic and the static criteria were implemented.

The initial temperature, t_0, is calculated by running a fast (rapid temperature decrease) schedule and picking-up the temperature for which more than 95% of the iterations are accepted. Temperature is usually decreased with a constant rate, α, usually close to one (*e.g.,* 0.90 or higher). Aarts an Korst (1990) showed that SA can find optimal solutions if equilibrium is attained at each temperature (constant objective function mean and variance) and proposed a temperature schedule dependent on the objective function variance that guaranties that. Despite this very attractive attribute such schedule tends to converge too slowly. Other t schedules for optimality were also proposed by Geman and Geman (1984), Hajek (1988), and Siarry (1997). These however may take too much time for many practical problems (Cohn and Fielding, 1999). The wealth of practical experience with the faster t schedule used here indicate that the solutions found should be *good* local optimal ones. In practical terms: good approximate solutions are a compromise between relatively good solutions in an amount of time significantly smaller than the one necessary to guaranty very good quality solutions provided (in theory) by slower schedules.

The generation mechanism of a new solution, U, from the neighborhood, $N(X)$ of a solution X is a simple one: Consider the solution vector $X=\{x_1, \ldots, x_\omega\}$; i) randomly remove one element (station) from X; ii) randomly choose an element in $\chi \setminus X$; iii) insert this element in X, obtaining U.

Simulated annealing has had many applications in water management studies and in related fields, and proved to be a good optimization algorithm for difficult combinatorial problems.

Examples are the works by Kuo *et al.* (2001), Kuo *et al.* (2003) and Berkowitz and Hansen (2001) in the optimization of crops irrigation; by Fallat and Dosso (1998) and Popov and He (2000) in geophysical data inversion for groundwater detection; by Skaggs *et al.* (2001a;b) and Johnson and Rogers (2000) for groundwater remediation optimization schemes; by Cunha and Sousa (1999) for water distribution network design; by Johnson and Rogers (2001), Wang and Zheng (1998), Cunha (1999), Dougherty and Marryott (1991), and Marryott *et al.* (1993) for groundwater management; and by Ferreyra *et al.* (2002), Brus *et al.* (2000), Dixon *et al.* (1999), Meyer *et al.* (1994), Storck *et al.* (1997), and Lee and Ellis (1996) for sampling optimization.

5.5 APPLICATION TO THE CASE-STUDY

A specific hydrochemical monitoring campaign was carried out by the Mining and Geological Institute of Portugal (IGM) to assess spatial and temporal variability of nitrate concentrations in the Beja`s Gabbro Aquifer System, in the vicinities of the city of Beja (Figure 5.1). Monitoring was undertaken between July 1997 and July 2000 in deep (30-40 m) and shallow (<10 m) wells and springs inside the aquifer, or draining from it, in a total of 89 points. In some of the times no record was possible either because wells or springs were dry, or water levels too deep. This is reflected in sparse data matrices. Nitrate concentrations fluctuate between 53 and 86 mg/l, though concentrations can easily exceed more than 100 mg/l in shallow wells, well above the maximum admissible value of 50 mg/l imposed by European norms (Table 5.1). In spite of all scientific evidences of diffuse pollution occurrences, the area was recently included in the irrigation perimeter of Alqueva Dam Project (the largest artificial lake in Europe). This has led to the need to establish a monitoring network with a relatively small, but representative number of stations, in which technical and human resources may be efficiently applied. The dimension of the groundwater nitrate monitoring network was asked by IGM to be kept below thirty stations concentrated in the most intensively used land area.

This area corresponds approximately to a rectangle with 9.0 x 5.5 km with longest axes in the SE-NW direction (Figure 5.1).

Table 5.1. Statistical parameters for the 19 measuring times (mg/l)

Measuring times	Ω	Mean	Min	q_{25}	q_{50}	q_{75}	Max	s^2
July 97	28	62.3	25.0	42.0	57.0	70.0	132.0	26.9
September 97	19	79.5	18.0	54.0	73.0	94.0	217.0	45.2
January 98	19	74.7	27.0	40.0	67.0	94.0	225.0	47.2
March 98	36	81.4	29.0	55.5	78.0	98.5	191.0	38.3
May 98	41	86.5	39.0	64.0	85.0	101.0	186.0	31.8
June 98	36	83.2	39.0	55.0	83.5	105.0	170.0	33.4
July 98	50	72.7	14.0	50.0	66.5	88.0	190.0	37.0
August 98	50	79.5	13.0	54.0	70.5	97.0	175.0	37.9
November 98	53	76.9	19.0	52	75.0	89.0	170.0	33.1
January 99	47	69.5	24.2	45.7	62.7	75.9	178.2	33.9
March 99	50	60.3	15.0	49.3	56.0	69.8	159.0	25.5
May 99	52	82.2	11.0	55.8	78.0	98.3	195.0	38.8
July 99	48	69.3	11.0	46.3	60.5	85.5	196.0	34.4
October 99	59	70.3	15.0	55.0	65.0	77.5	150.0	25.6
November 99	63	76.9	30.0	50.0	70.0	89.5	156.0	31.7
January 2000	65	83.4	10.0	61.0	78.0	95.0	178.0	32.0
March 2000	65	96.7	32.0	69.0	88.0	111.0	191.0	37.3
May 2000	59	92.8	28.0	70.5	84.0	116.0	188.0	36.2
July 2000	67	88.7	13.0	70.5	87.0	104.0	162.0	29.4

5.5.1 Selecting the optimal monitoring network

Data from the existing monitoring network is very incomplete both in the number of stations measured at each measuring time and in periodicity. Moreover records extend for only three years. With these restrictions in mind a small GMN, with less than 30 stations, is to be selected for subsequent monitoring. The monitoring interval was specified by IGM to be three months and was not subjected to optimization.

Due to the sparsity of time-space data matrix and to the small number of measuring times (only 19) it was decided to proceed with the spatial autocovariance analysis for each measuring times, of which resulted the spherical variogram models presented in Table 5.2. Spherical

variogram models are analytical expressions adjusted to the experimental variograms, having an equation of the form,

$$\gamma(h) = C_0 + \begin{cases} C \cdot \left[1.5\frac{h}{r} - 0.5\left(\frac{h}{r}\right)^3 \right] & if\ h \le r \\ C & if\ h \ge r \end{cases} \qquad (5.9)$$

where C_0 is the small scale variability due to analytical errors or to variability associated with structures with dimension inferior to the spacing between data locations; C is the amount of correlated variability; r is the distance after which there is no evident spatial autocorrelation (the experimental variogram is flat or oscillates around a constant value).

Table 5.2 also shows the number of stations available, identified by Ω, for the 19 monitoring times. Clearly this value varies from one measuring time to another, with a minimum of 19 stations in September 1997 and a maximum of 67 stations in July 2000.

If these results were taken as are, nineteen optimization procedures would have to be made according to the methodology proposed earlier, thus leading to a very long optimization process. Moreover redundant results are expected due to correlation between different measuring times. As optimization is based on the estimation variance and it depends on the variogram model parameters, it seems reasonable to use variogram information. The selection of a lower number of measuring times is made by picking the times further apart in a principal components space constructed with the variogram models parameters. The factors (axes in a graphic projection) resultant from principal components analysis (PCA) are independent and reflect different causes of variation. Hence two points significantly apart in a graphical PCA projection also have significantly different characteristics. The two factors of principal component analysis on the variogram parameters are shown in Figure 5.3. They explain 84% of the total variability (54.2% for the first and 28.8% for the second). The first factor is more

strongly related with the sill of the variogram, and the second is more strongly related with the nugget effect. The range of the variogram is equally related with all PCA factors.

Eight measuring times were selected out of the original nineteen according to their relative distance in this principal components space, first by selecting those further apart in the first factor and then those further apart in the second: {March 1998, May 1998, November 1998, March 1999, May 1999, October, 1999, March 2000, July 2000}. The number of measuring times was chosen such that the processing time was kept reasonably low, and such that the main types of variogram models were included.

Table 5.2. Parameters of spherical space variograms models for the 19 measuring times

Measuring times	Ω	C_0 (mg/l)²	C (mg/l)²	r (m)
Jul. 97	28	345.0	700.0	3000.0
Sept. 97	19	962.0	1931.4	2762.0
Jan. 98	19	609.0	2107.0	4336.5
Mar. 98	36	975.0	1440.0	1404.0
May 98	41	742.5	987.4	2538.0
Jun. 98	36	675.0	1084.5	1969.0
Jul. 98	50	862.0	1341.5	3450.0
Aug. 98	50	865.0	1405.0	3342.0
Nov. 98	53	836.8	1073.7	994.0
Jan. 99	47	849.2	1124.2	2887.0
Mar. 99	50	540.1	637.6	2020.0
May 99	52	317.3	1477.6	1977.0
Jul. 99	48	530.8	1156.4	1870.0
Oct. 99	59	270.6	644.9	1700.0
Nov. 99	63	715.8	986.1	1240.0
Jan. 2000	65	487.4	1007.5	2400.0
Mar. 2000	65	308.0	1368.7	1177.0
May 2000	59	250.0	1289.3	3000.0
Jul. 2000	67	393.6	850.9	1427.0

A question remains to be answered: how many stations should the new GMN have in order to guaranty that it can be used to detect, in average, the non-compliance to the concentration

standard? According to IGM an error in the estimation of the average nitrate concentration between 10 and 15 mg/l is acceptable, due to the already high concentrations detected in the region (these values are also very close to the admissible error established by EU legislation in 12.5 mg/l (EU, 1998)). With these values, which we will denote as d, and data variance, s^2, using the measuring time with highest data variance, it is possible to estimate the number of stations in the new design necessary to estimate the mean with an error of d for a given confidence level, α_L by (Cochran, 1977)

$$\omega = \frac{1}{\frac{d^2}{z^2 s^2} + \frac{1}{\Omega}} \qquad (5.10)$$

where z denotes the upper $\alpha_L/2$ point of the standard normal distribution. The variance for May 1999 equals 1506.5 (mg/l)^2. Assuming a small probability that d exceeds the established values, say 10% (α_L=0,10), z comes equal to 1.645. If d = 10 mg/l, the number stations should be ω= 23, and if d = 15 mg/l, ω= 13. An intermediate ω value of 18 stations was chosen.

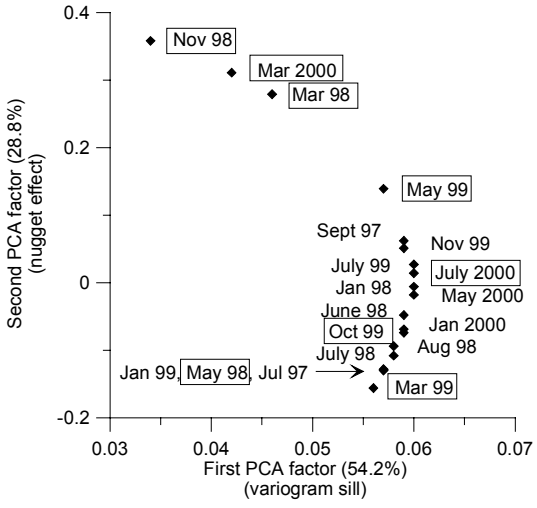


Figure 5.3. Projection of measuring times in the first two PCA factors of variogram parameters. Selected measuring times are identified

A *first-approximation GMN* dimension of 18 stations for each measuring time was optimized with simulated annealing by letting the algorithm find the set of 18 stations out of the original sets with the lowest σ_E^2 value. The following annealing parameters were used: 100Ω iterations or 10Ω successful solutions (variable for each measuring time because Ω also varies); α=0.9; R_{STOP}=3. The eight sets of results are shown in Table 5.3. In the table are also included the number of combinations of 18 stations out of Ω original stations (feasible space). It is evident that the number of combinations is always enormous and trying to solve the combinatorial problem exhaustively is an impossible task.

Objective function values, time necessary to attain a solution, *T*, and number of iterations show some correlation with the number of available stations, Ω, and reflect the same characteristics: as more stations are available more combinations of stations are available and SA may look for better solutions, but at the expenses of testing more solutions and taking more time. Spatial autocovariance seems to play a minor part when it comes to differentiate combinatorial problems' structure (complexity of the hipper-dimensional objective function defined by all the possible combinations of stations). Also no relation is evident between variogram parameters, simulation time *T,* and initial annealing temperature t_0 (Table 5.3 and Figure 5.4). This does not mean that the optimization results are independent of variogram models, of which the objective function depends, rather that the problems combinatorial structure (mainly number and proximity of the objective function minima) does not clearly show this relation. Figure 5.4 also shows that convergence to a minimum σ_E^2 value is very fast at high temperature values, very low at temperatures close to zero, stabilizing finally at a minimum optimal value. This behavior reflects one of SA characteristics: at high temperatures most of the candidate solutions are accepted, σ_E^2 value is still high and may even increase between consecutive temperatures; as temperature decreases more successful solutions are

accepted and σ_E^2 only decreases; finally at temperatures close to zero only successful solutions are accepted and the algorithm stops when σ_E^2 can no longer be lowered.

Table 5.3. Optimization results

Measuring time	Iterations	Time (T) (s)	σ_E^2 $(mg/l)^2$	t_0	Combinations Log_{10}
Mar. 98	22737	1578	$8.49x10^{-2}$	0.40	9.96
May 98	156078	10834	$3.16\ x10^{-2}$	0.40	11.31
Nov. 98	98318	6804	$3.44\ x10^{-2}$	0.06	13.81
Mar. 99	116038	8050	$6.74\ x10^{-3}$	0.03	13.26
May 99	146567	10195	$3.32\ x10^{-2}$	0.40	13.63
Oct. 99	157865	10977	$6.51\ x10^{-3}$	0.10	14.81
Mar. 2000	171975	11927	$1.20\ x10^{-2}$	0.60	15.70
Jul. 2000	20675	1436	$7.43\ x10^{-2}$	0.45	9.96

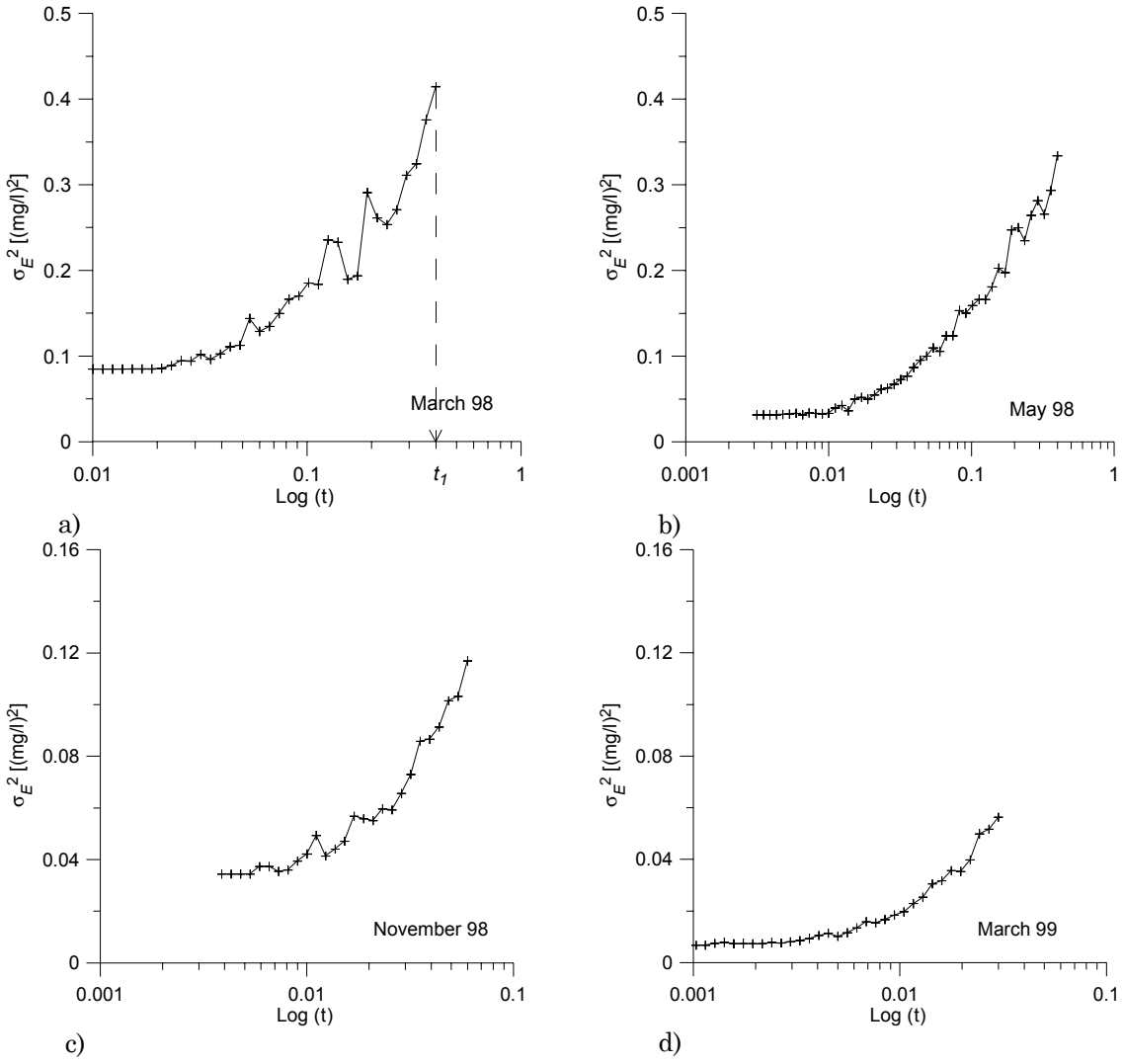


Figure 5.4. Objective function convergence curves (x10³). Temperature in log scale.

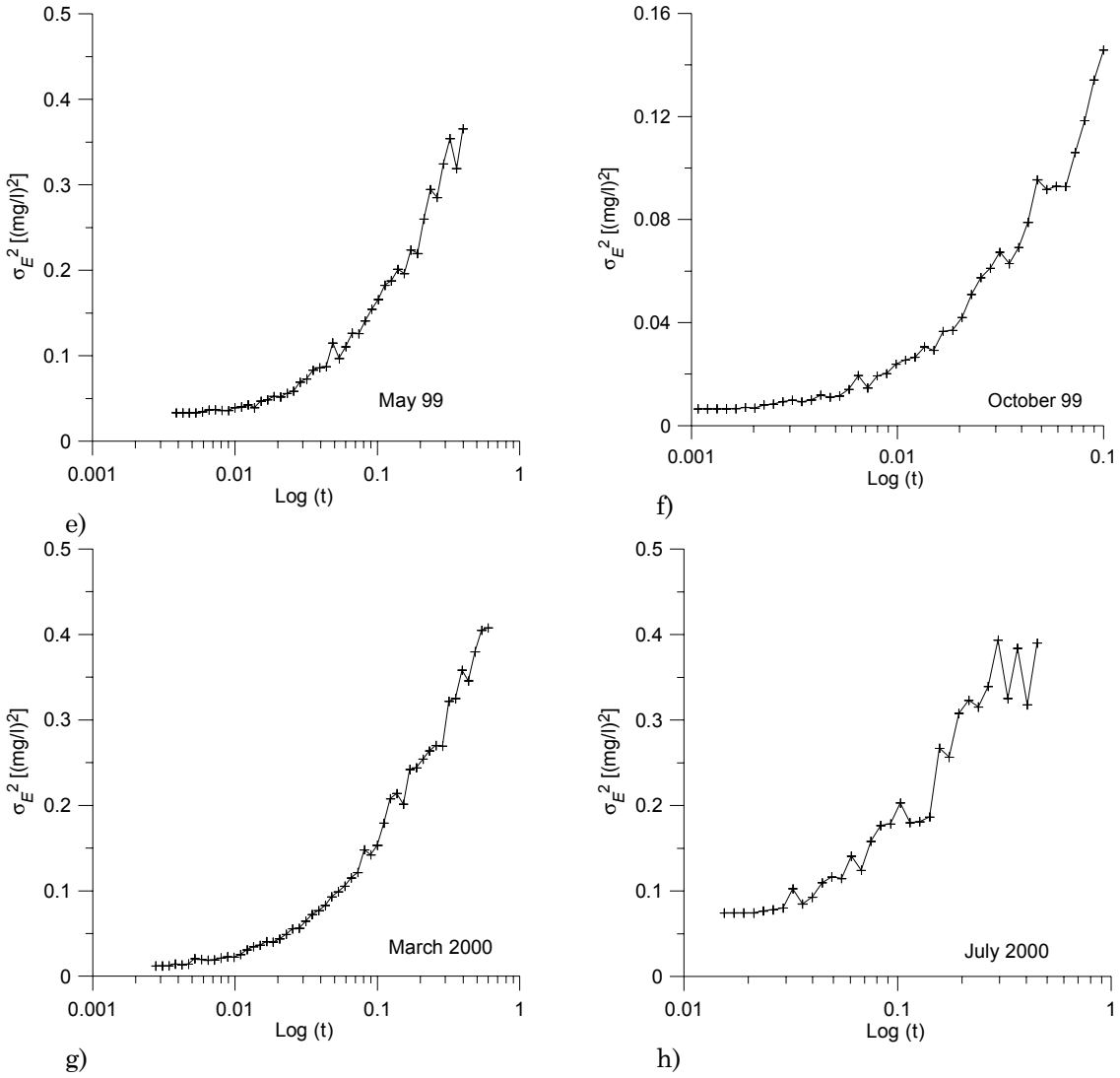


Figure 5.4. Objective function convergence curves (x10^3). Temperature in log scale. (cont.)

Nitrate concentrations fields estimated with the variogram parameters of Table 5.2 are shown in Figure 5.5. Figure 5.6 shows the optimal spatial distributions of stations (circumferences) along with the available stations for each measuring time (plus signs). With isotropic variogram models, in theory and for slowly varying concentration fields, an optimal spatial distribution is one with stations more or less evenly distributed in space. However, if the concentration fields are not smooth, important estimation errors may appear in stations where values are quite different from their neighbors. This may imply that in some cases the inclusion of more stations in high variability areas results in a more clustered distribution, but

with lower estimation error variance. This is clearly seen by comparing the information in Figure 5.5 and Figure 5.6. Take for instance March 1999: The optimal distribution of stations is along the NW-SE diagonal such that the high variability found in that area may be well depicted. Similar interpretations can be made for the remaining measuring times.

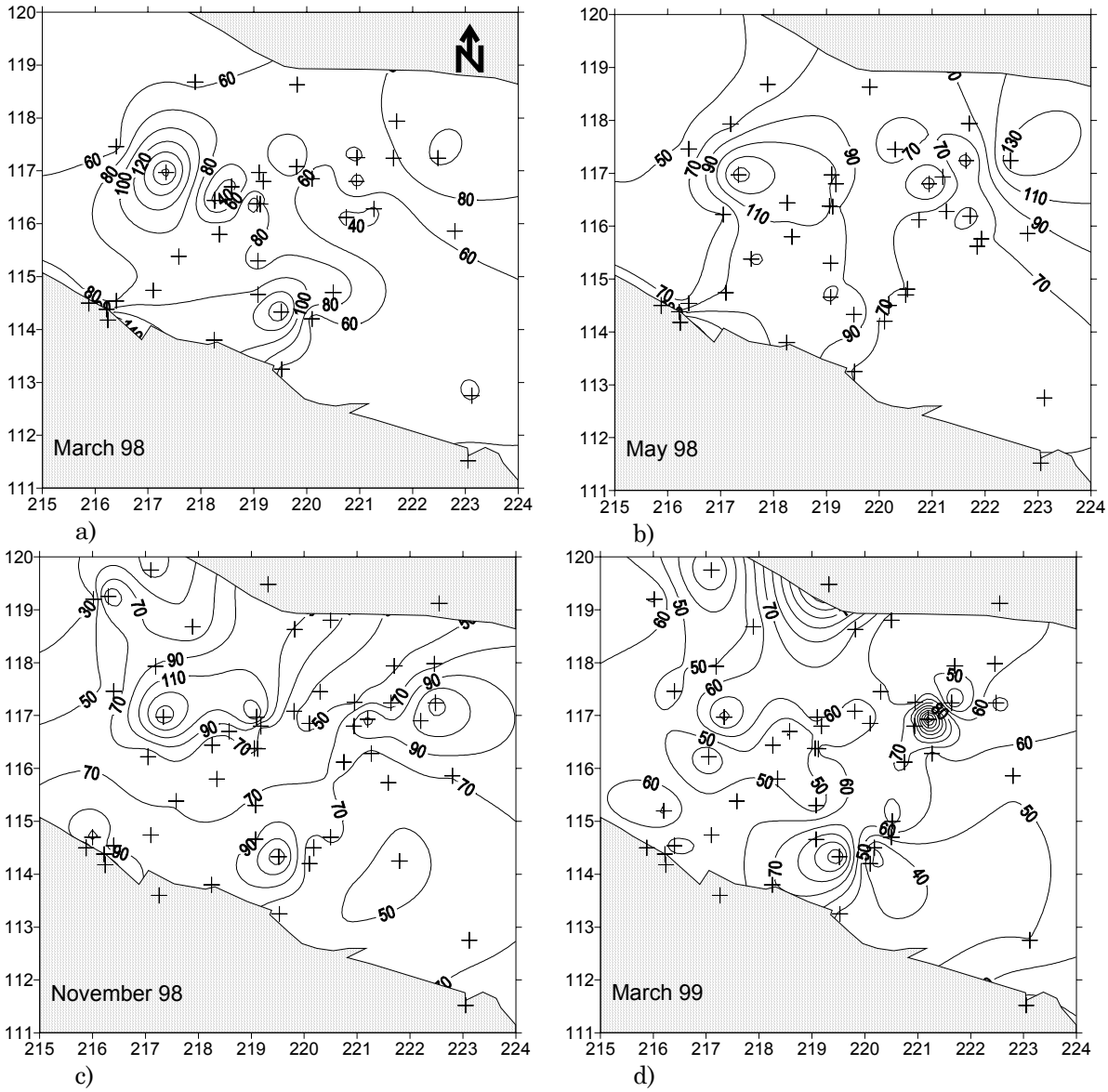


Figure 5.5. Nitrate concentrations (mg/l). Original set of stations represented by plus signs. X and Y in Portuguese military coordinates (x 10^3 m)

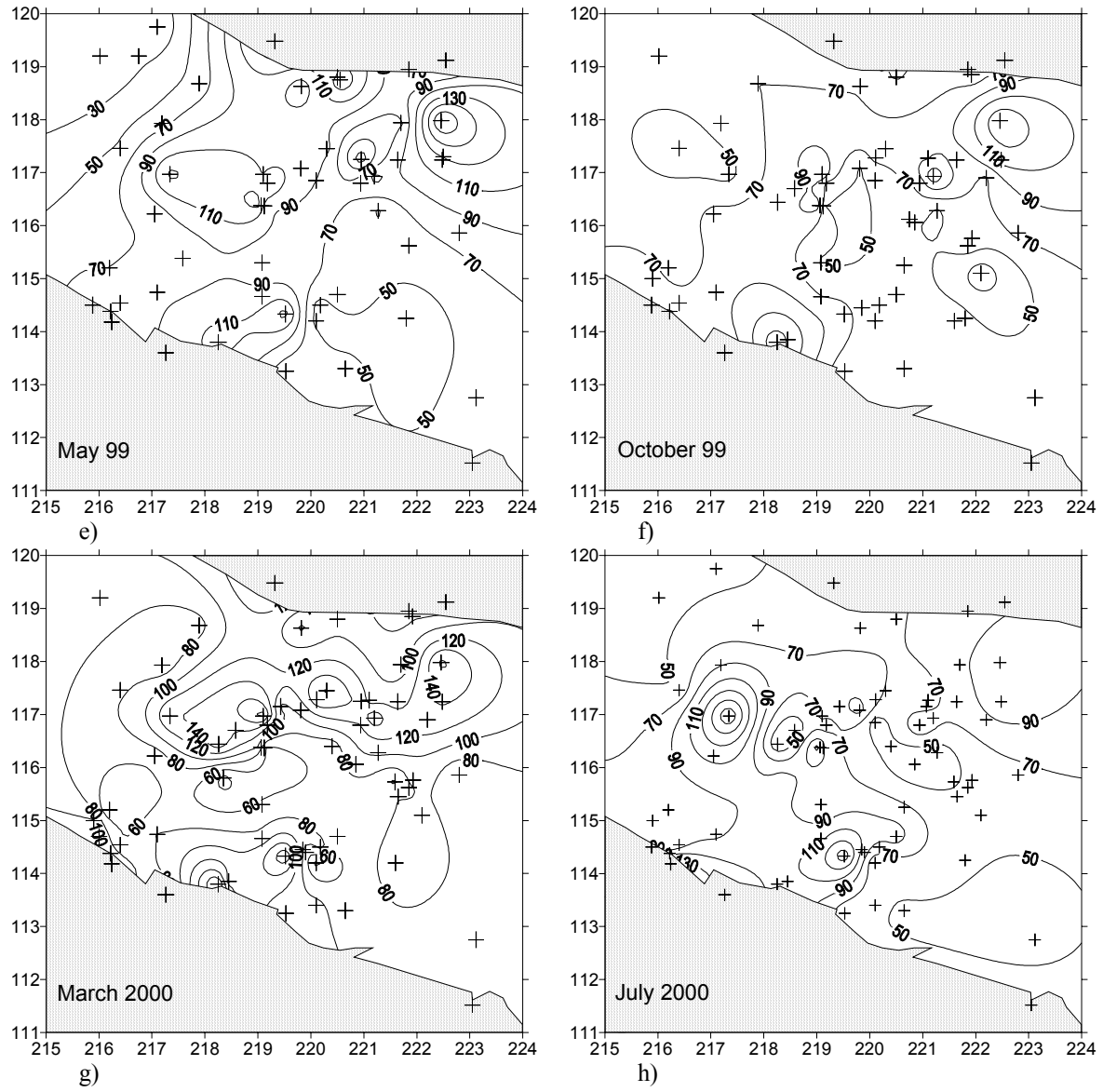


Figure 5.5. Nitrate concentrations (mg/l). Original set of stations represented by plus signs. X and Y in Portuguese military coordinates (x 10^3 m) (cont.)

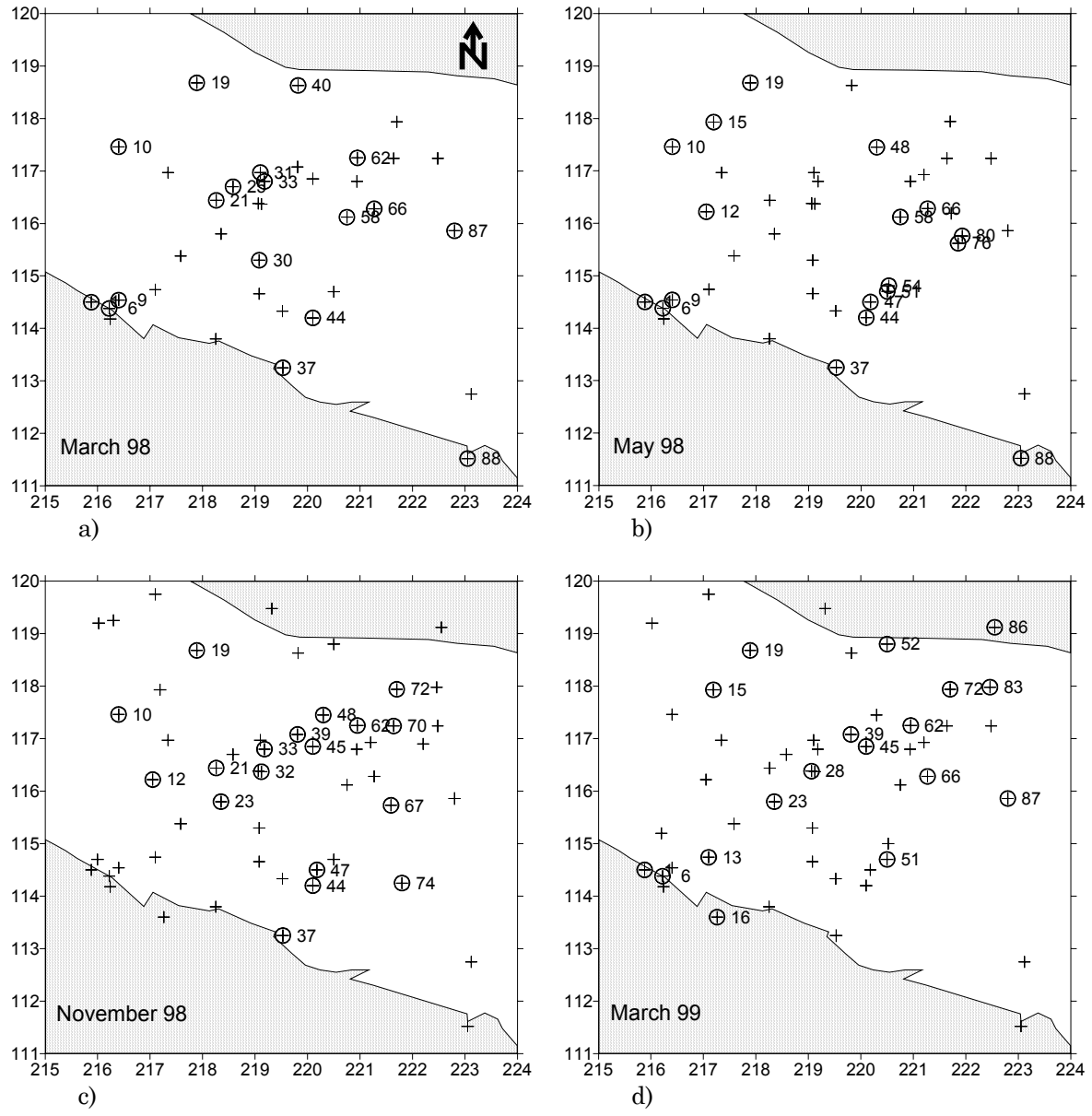


Figure 5.6. Optimal spatial distribution of stations in the eight measuring times. Original set of stations represented by plus signs and optimal first approximation network by circles. X and Y in Portuguese military coordinates (x 10^3 m)

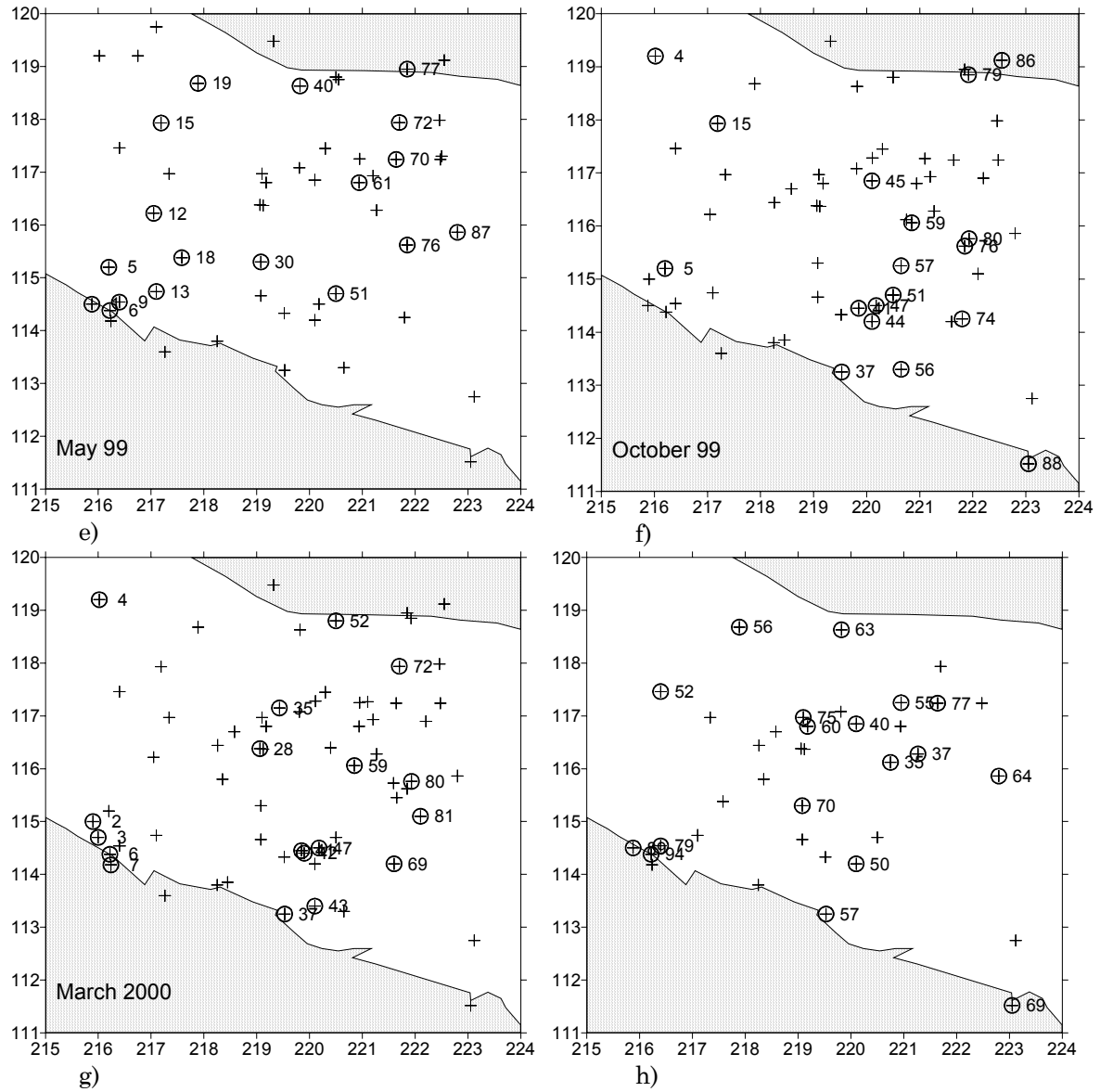


Figure 5.6. Optimal spatial distribution of stations in the eight measuring times. Original set of stations represented by plus signs and optimal first approximation network by circles. X and Y in Portuguese military coordinates (x 10^3 m) (cont.)

In order to obtain a unique set of stations a frequency analysis is performed by computing the number of times a single station occurs in the eight measuring times. Out of the original set of eighty nine stations, thirty-one were never chosen, twenty were only chosen once, fourteen were only chosen twice, eight were chosen three times, eleven were chosen four times, two were chosen five times and three were chosen six times.

In order to decide if one station enters the new design the following simple rule was used: i) calculate the frequency in which each station occurs in the optimized *first approximation GMN* networks, *i.e.*, in all times; ii) choose the stations with higher frequencies. Two candidate sets of stations were chosen, one in which stations appeared at least three times, and another where frequency was at least four. The first set contains 24 stations (Figure 5.7a) and the second 16 (Figure 5.7b). These were the *final GMN* that were tested, which have dimensions close to the ones estimated with equation (5.10)

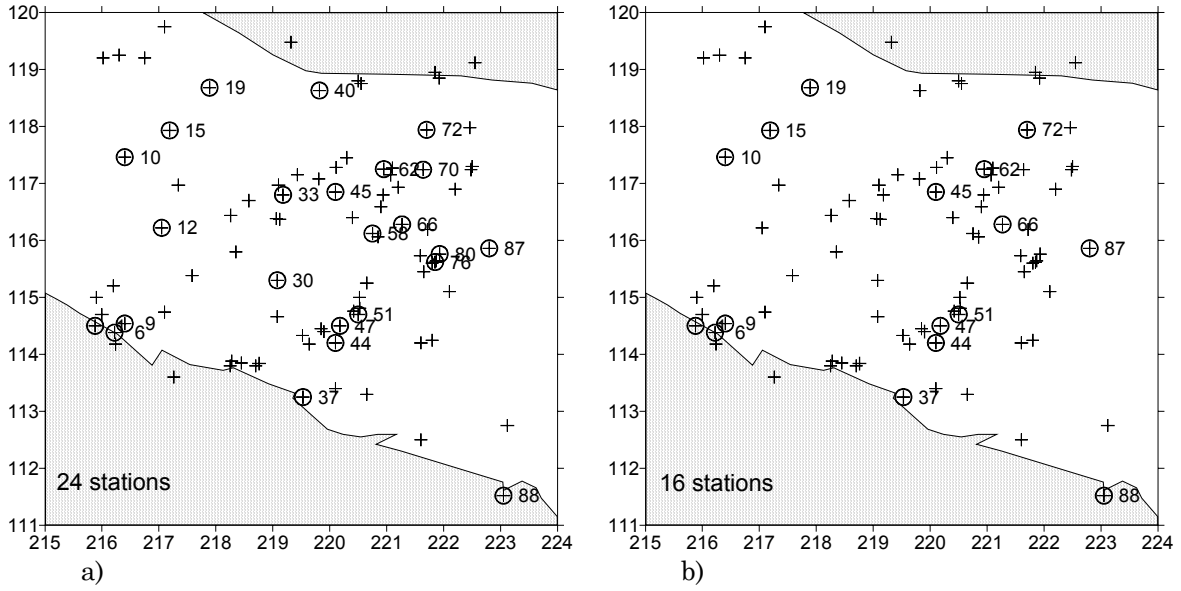


Figure 5.7. Stations included in the GMN. With frequency of at least a) three stations; b) four stations. Original set of stations represented by plus signs and optimal networks by circles. X and Y in Portuguese military coordinates (x 10^3 m)

5.5.2 Validation of selected monitoring networks

In order to validate the selected GMNs two measuring times are chosen, one belonging to the set used to obtain the candidate solutions (July 2000), and another that was not used (November 1999). These are the only two measuring times with records in the majority of the

stations of the *final GMN*s (in November 1999 station 62 is missing and in July 2000 stations 58 and 62 are missing).

The validation proceeds by i) estimating a field of 9.0 x 9.0 km with the original data set; ii) estimate again using only the stations in the candidate GMNs; iii) compute absolute relative errors and their mean (equations (5.11) and (5.12)).

$$|\varepsilon_i| = \left|\left(z^*(x_i) - z^{**}(x_i)\right)/ z^*(x_i)\right| \quad (5.11)$$

$$\overline{|\varepsilon|} = \frac{1}{B}\sum_{i=1}^{B}|\varepsilon_i| \quad (5.12)$$

B represents the number of blocks used to discritise the estimated field, $z^*(x_i)$ and $z^{**}(x_i)$ stand for the estimated mean value in each block with the original data set of stations, and with the optimal *final GMN*, respectively. Equation (5.11) quantifies the local amount of error obtained by either over- or underestimating, and equation (5.12) the mean error over the entire field.

The same variogram models are used in each measuring time to estimate by ordinary kriging a mesh of 200 x 200 m with the original GMN and with the *final GMN*. Estimated values outside the borders of the aquifer are discarded.

The *final GMN*s when used to re-estimate July 2000 nitrate concentrations result in mean relative absolute errors lower than those for November 1999 (Table 5.4), indicating that the *final GMN*s interpolations have lower mean relative absolute errors than the extrapolations, which is an expected result.

Table 5.4. Mean relative absolute error (%)

	GMN dimension	
Measuring time	24	16
July 2000	6.13	6.95
November 1999	8.50	9.26

Figure 5.8a shows the estimated nitrate concentrations for November 1999 using all available stations and the variogram parameters of Table 5.2. The same information for July 2000 is in Figure 5.5h. Estimated fields for these months show many resemblances, with high and low values in the same locations. This feature is indeed common to most of the studied measuring times.

The twenty-four stations *final GMN* and in July 2000, despite a mean relative absolute error of 6.13%, shows some localized areas with $|\varepsilon_i|$ passing 20%, namely in the central and SW part of the field (Figure 5.8c). The great majority of the stations from the original data set are in areas with $|\varepsilon_i|$ lower than 10%, which is also reflected in the low mean error. A very similar interpretation is made to November 1999 (Figure 5.8b), but $\overline{|\varepsilon|}$ is slightly higher (8.50%). This *final GMN* would promote a reduction of approximately 70% in the monitoring costs.

For the sixteen stations *final GMN* the resulting $|\varepsilon_i|$ values are slightly higher, with some of the stations laying in areas with $|\varepsilon_i|$ up to 25% (Figure 5.8d and e). However, these stations correspond to locations with very high nitrate concentrations, above 70 mg/l, indicating that an error of plus or minus 25% in the estimated value would still indicate non-compliance to the 50 mg/l limit value. This network dimension not only fulfils the monitoring purpose, but also implies a reduction in exploitation costs of near 80% when compared to the original network, and of 25% when compared to the other candidate GMN.

These results were presented to IGM that decided to implement the *final GMN* with 16 stations.

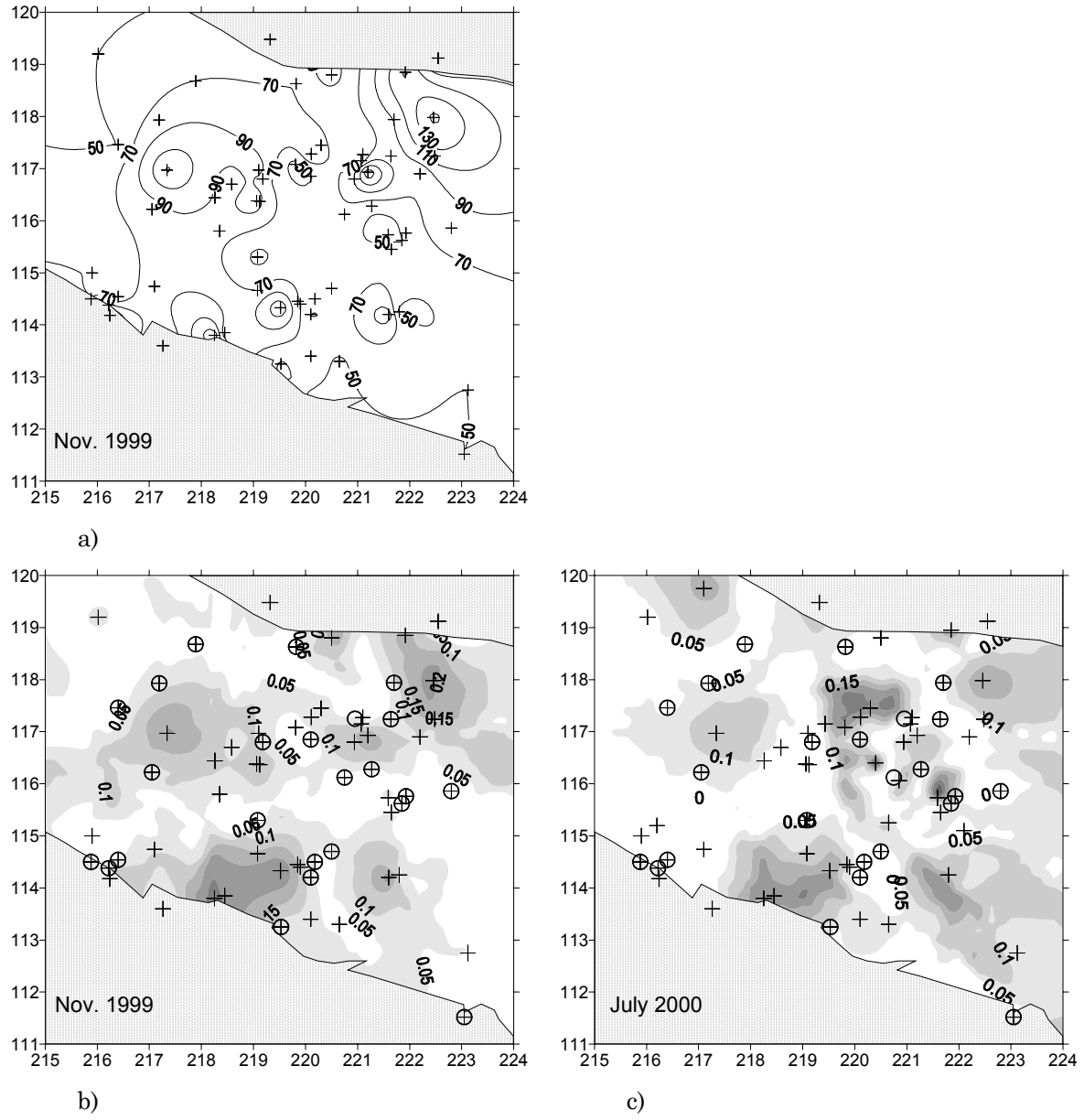


Figure 5.8. Field estimated with raw data for a) Nov. 1999; Mean relative absolute errors with 24 stations for b) Nov. 1999; c) July 2000; idem with 16 stations for d) Nov 1999; e) July 2000. Original set of stations represented by plus signs and optimal networks by circles. X and Y in Portuguese military coordinates (x 10^3 m)

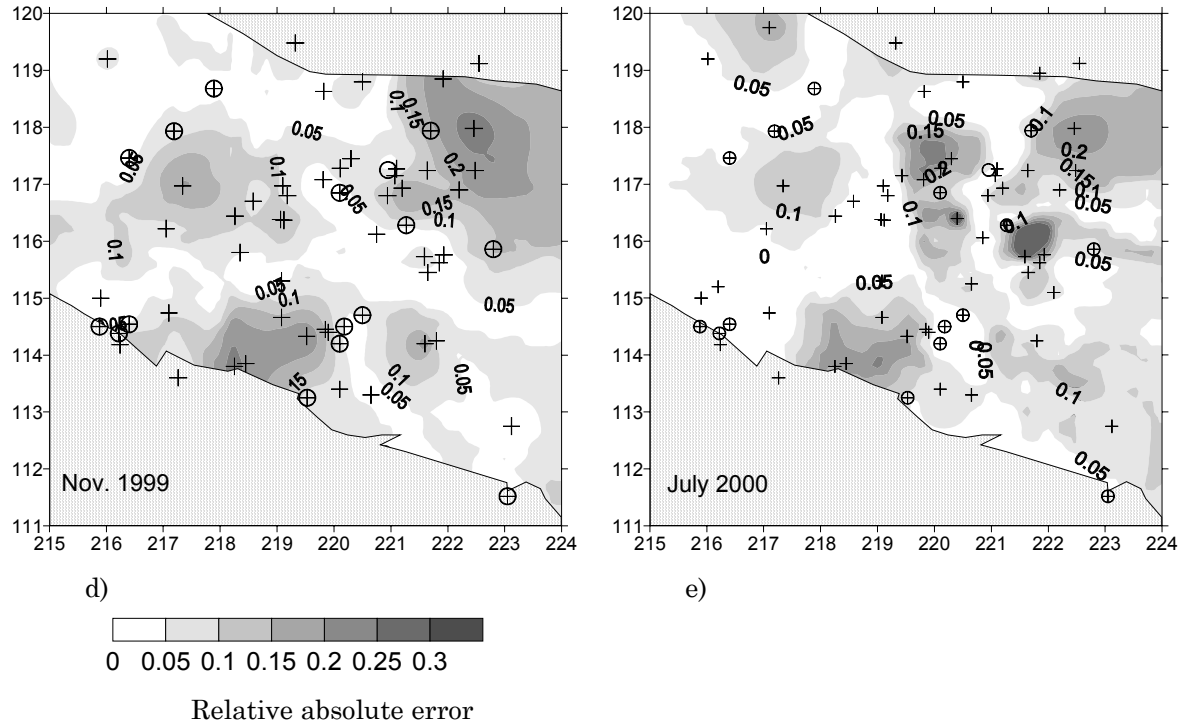


Figure 5.8. Field estimated with raw data for a) Nov. 1999; Mean relative absolute errors with 24 stations for b) Nov. 1999; c) July 2000; idem with 16 stations for d) Nov 1999; e) July 2000. Original set of stations represented by plus signs and optimal networks by circles. X and Y in Portuguese military coordinates (x 10^3 m) (cont.)

5.6 CONCLUSIONS

When time series from an existing groundwater reference level monitoring network are too incomplete and a small number of representative stations has to be selected from the original set, difficulties appear due to i) the same stations are not monitored at the same measuring times; ii) if one single measuring time is used the resulting optimized network may miss detecting features present in other measuring times (because of, *e.g.*, seasonality in precipitation and fertilization or trends, such as droughts); and iii) spatial autocorrelation changes with time (due again to seasonality or change of land use). A method that allows dealing with these difficulties was proposed and applied to a case-study.

If the time series are too incomplete, some of the selected reduced-dimension monitoring stations may be considered as optimal locations in a small fraction of the measuring times used to optimize the network, but are nevertheless important to identify locations with extreme seasonal values. The method allows selecting a monitoring network that captures the spatial variability identified with the large-dimension original monitoring network. A reduction of a groundwater monitoring network with 89 to one with 16 or 24 stations is proposed, implying a reduction in monitoring costs between 70% and 80%. The proposed groundwater monitoring networks introduce average relative absolute errors in the estimated spatial distributions of nitrate concentrations that are less than 10% when compared to the field estimated with the complete data set. In some locations, where nitrate concentrations are high (usually above 70 mg/l), the local relative absolute error introduced is as high as 25% (also the maximum admissible error imposed by European legislation (EU, 1998), *i.e.*, the estimated concentrations may be over or under-estimated by this amount. However, even with such error, regulatory compliance at the 50 mg/l can be detected. Groundwater monitoring network validation from time to time is a necessary procedure to verify its efficiency. Validation should be made sooner if nitrogen inputs change or if groundwater flow regime is altered.

Although presented within a specific case study, the method is easily adapted to different conditions. For example, if the number of measuring times is too large then the raw data may be used instead of using variogram parameters to reduce the number of measurement times included in the analysis. The main difficulty is that with more PCA degrees of freedom the eigenvalues with each factor are low and the selection of representative measuring times becomes more difficult.

6 OPTIMAL ESTUARINE SEDIMENT MONITORING NETWORK DESIGN WITH SIMULATED ANNEALING

6.1 SYLLABUS

A geostatistical variance reduction based objective function, constrained to the reproduction of the probability distribution functions of selected sediment physic-chemical variables, is applied to the selection of the best set of compliance monitoring stations in the Sado river estuary in Portugal. The stations were to be selected from a large set of sampling stations from a prior field campaign. Simulated annealing was chosen to solve the optimization function model. Both the combinatorial problem structure and the resulting candidate sediment monitoring networks are discussed, and the optimal dimension and spatial distribution is proposed. An optimal network of sixty stations was obtained from an original 153 stations sampling campaign.

6.2 INTRODUCTION

In many monitoring programs a first sampling stage with a large number of locations is made, either because there is no prior information, or it is considered necessary to collect more data. This stage is usually planned to give statistical information about the variables under study and to calculate their spatial covariance. A second stage is needed to transform the original set of sampling stations, with high cardinality, into a lower cardinality set of monitoring stations. Probably the most used methods to reduce the cardinality are those based on the maximization of the spatial accuracy, or put in other words, on the minimization of the variance of the estimation error, also known as variance reduction methods. This is usually made in the context of geostatistical theory (Matheron, 1963; Matheron, 1965) and most frequently by interpolation with unknown mean, *i.e.*, by ordinary kriging.

A particular case of variance reduction methods, the fictitious point method (Delhomme and Delfiner, 1973), was applied here to select the number and position of sediment monitoring stations in the Sado river estuary in Portugal, such that different physic-chemical homogeneous areas identified in a prior sampling campaign were considered. This monitoring network will be further integrated on an environmental management system for Sado Estuary as a decision support tool for local authorities: Sado Estuary Natural Reserve and Setúbal Port Administration.

For practical and budgetary reasons the number of monitoring stations should be reduced to a minimum. The optimization problem can be stated in a very simple way by: maximizing the spatial accuracy, constrained to a maximum number of stations, given the information collected in a prior sampling program (153 sampling sites). Maximize the spatial accuracy is easily attained by minimizing the variance of estimation error, while incorporating the

patchiness of homogeneous areas is a more difficult problem. One alternative would be to fix several locations inside the different homogeneous areas, but then the choice of stations would be arbitrary. Another way is by stratification, considering that a defined number of stations must be placed inside homogeneous areas. Stratification is a well-known statistical technique intended to design monitoring (or sampling) programs with denser networks in some areas than in others. The difference in probability density may be based on, *e.g.*, spatial autocovariances, statistical risk of contamination, plume detection probabilities, empirical judgment, among many others. Here we propose a statistically based stratification: homogeneous areas are monitored according to the frequency in which they appear in the prior sampling program.

The article is divided in five sections. This Introduction, a second section where the theoretical geostatistical and optimization framework is presented. In this section the most frequently geostatistical parameter used to measure accuracy, the kriging estimation error variance, is explained and compared with another geostatistical measure of accuracy, the fictitious point estimation error variance. Also the simulated annealing heuristic used to solve the optimization problem is introduced. In the third section the case-study is presented and data transformations explained. Optimization results are discussed in the fourth section. Finally the most important conclusions are drawn in the last section.

6.3 THEORY

6.3.1 Estimation of probability distribution functions

Indicator coding implies transforming a continuous or discrete variable, $Z(x)$, into a discrete (0,1) one, the indicator, $I(x)$. Considering a threshold value z_c on Z, $I(x)$ is equal to 1 if $Z(x) \leq z_c$,

and 0 otherwise. Therefore the variable at each location is transformed into a distribution function, *i.e.*, the probability of exceeding the threshold is calculated within a region. With a sufficiently large number of thresholds the prior (and post) probability distribution of Z is calculated at each location. Indicator transform is also in the core of nonparametric methods, which have some clear advantages over parametric methods: i) the parametric hypothesis may not hold; ii) there are no statistical tests to investigate adequately investigate the validity of a multivariate distribution hypothesis (Alli *et al.*, 1990); iii) parametric methods are difficult to comprehend and apply by many practitioners due to their mathematical complexity (Sullivan, 1984). Several geostatistical methods can be used to estimate probability distributions, namely Multigaussian kriging (MK), disjunctive kriging (DK), lognormal kriging (LK), probability kriging (PK) and indicator kriging (IK). Both MK and DK are based on normality assumptions, and LK on log-normality assumption. If these assumptions are not verified, *e.g.*, in highly skewed distributions, variogram fitting tends to be very problematic, and the estimation of poor quality. IK is therefore a good alternative. It is actually one of the most frequently used nonparametric methods and will also be used here.

The theory and implementation of nonparametric estimators of spatial distributions is similar to that of nonparametric estimators of the local mean (Journel, 1983). The following paragraph follows the aforementioned author.

Consider the indicator

$$i(x, z_c) = \begin{cases} 1, & if\ z(x) \le z_c \\ 0, & otherwise \end{cases} \qquad (6.1)$$

The spatial distribution of a random variable on a point support within a region A is

$$\phi(A, z_c) = \frac{1}{A} \int_{x \in A} i(x, z_c) dx \qquad (6.2)$$

The spatial distribution (6.2) can be considered a realization of the following random variable

$$\Phi(A, z_c) = \frac{1}{A} \int_{x \in A} I(x, z_c) dx \qquad (6.3)$$

The expected value of this random variable is (for stationarity of *z(x)*)

$$\begin{aligned} E[\Phi(A, z_c)] &= \frac{1}{A} E\left[\int_{x \in A} I(x, z_c)\, dx\right] = \frac{1}{A} \int_{x \in A} E[I(x, z_c)] = \\ &= \frac{1}{A} \int_{x \in A} \{(1) \Pr[z(x) \le z_c] + (0) \Pr[z(x) > z_c]\} dx = \Pr[z(x) \le z_c] = F(z_c) \end{aligned} \qquad (6.4)$$

$F(z_c)$ is the stationary univariate cumulative distribution function. The purpose of indicator transformation is to estimate the cumulative probability function (posterior) (equation (6.2)). The $\phi^*(A,z_c)$ functions are linear combinations of the indicator function and represent the proportion of values less than the threshold,

$$\phi^*(A, z_c) = \sum_{\alpha=1}^{\omega} \lambda_\alpha \, i(x, z_c) \qquad (6.5)$$

λ_α are weights, with the constraint $\sum_{\alpha=1}^{\omega} \lambda_\alpha = 1$ for unbiasedness and α the number of stations.

Equation (6.5) can be solved by simple kriging using $i(x,z_c)$ and indicator variogram:

$$\gamma_i(h, z_c) = \frac{1}{2N_h} \sum_{i=1}^{N_h} [i(x + h, z_c) - i(x, z_c)]^2 \qquad (6.6)$$

N_h is the number of pairs for lag h.

The estimator of (6.5) is, in the final form,

$$\phi^*(A, z_c) = \sum_{\alpha=1}^{n} \lambda_\alpha \, i(x, z_c)\, F(z_c)\,(1 - \sum_{\alpha=1}^{\omega} \lambda_\alpha) \qquad (6.7)$$

Also, if one is interested in the proportions associated with each interval, $]z_{c-1}, z_c]$, then,

$$\Pr[z_{c-1} < z(x_\alpha) \le z_c] = \phi(A, z_c) - \phi(A, z_{c-1}) = \Psi(A, z_c) \qquad (6.8)$$

and, for the estimated proportions

$$\phi^*(A, z_c) - \phi^*(A, z_{c-1}) = \Psi^*(A, z_c) \qquad (6.9)$$

6.3.2 Estimation error and estimation error variance

In this article the estimation error is looked for at the x_α locations because the monitoring stations to be included in the new design are taken out of the initial locations. The estimation error obtained by removing one of the x_α stations, estimate it by kriging with the remaining stations (those in the vicinity) and repeat this for all stations is also the heart of a geostatistical technique caled cross-validation (Deutsch and Journel, 1992). The estimation error obtained here in a similar way will be coined mean squared estimation error (mse), and is calculated by equation (6.10),

$$s^2 = \frac{1}{\omega - 1} \sum_{\alpha=1}^{\omega} \left[i(x_\alpha, z_c) - i^*(x_\alpha, z_c) \right]^2 \qquad (6.10)$$

where $i^*(x_\alpha, z_c)$ is the IK estimated value. Each station's value is obtained by removing it from the set and estimating it by IK using the remaining stations.

The spatial arrangement of points that minimizes s^2 has the lowest estimation error and therefore best reflects the spatial correlation introduced in the variogram model. By minimizing s^2 the stations that can best be predicted by the remaining stations are excluded. Mean squared estimation error uses both data and estimated values at points with known value to calculate the errors, reflecting the real influence of adding or removing stations on the estimation errors of the remaining (Delhomme, 1978). It will be used as measure of accuracy in the following discussion.

6.3.3 Optimization function model

The stations' locations that produce the lowest mse result in a spatial distribution with the highest accuracy. Therefore one wishes is to minimize equation (6.10). The optimization function model comes, considering a set, S, of all the original stations, with cardinality Ω, take a subset, S', with cardinality ω, such that $\omega < \Omega$.

Minimize

$$s_p^2 = \frac{1}{\omega - 1} \sum_{\alpha=1}^{\omega} \left[i(x_\alpha, z_c) - i^*(x_\alpha, z_c) \right]^2, \omega \in S', S' \subset S \qquad (6.11)$$

Subject to

$$\Psi_{S'}(A, z_c) \approx \Psi_S(A, z_c)$$
$$\varpi \leq \varpi_{Max}$$

The first constraint makes it necessary for a candidate solution set, S', to have the same proportion of stations with values in the intervals $]z_{c-1}, z_c]$ as the original set, S. The condition is not equality because for practical computation floating-point variables equality is machine dependent and varies with the precision. Instead, $\Psi_S(A,z_c)$ may be bounded, and the constraint becomes:

$$\Psi_S(A, z_c)(1 - \delta) \leq \Psi_{S'}(A, z_c) \leq \Psi_S(A, z_c)(1 + \delta) \qquad (6.12)$$

with δ the semi-amplitude of a]0,1] interval.

This condition is necessary to correct the bias introduced by variogram models fitting errors (when adjusting the theoretical models to the experimental variogram). A practical example showing the bias is presented in this article.

The second constraint limits the maximum number of stations to be included in the design.

6.3.4 Solving the optimization function model

The example studied here may be classified as a difficult combinatorial optimization problem, for which an exhaustive search of all possible combinations is not possible in a reasonable amount of time. Solutions to these problems may however be approached by heuristic algorithms that iteratively look for better solutions by trial and error. One of such algorithms is the well-known simulated annealing (SA). It is one of the threshold algorithms included in the class of local search algorithms. The other two are, as defined by Aarts and Korst (1990): iterative improvement, where only OF-reducing neighbors are accepted, and threshold accepting, where some deterministic non-increasing threshold sequence is used, allowing neighbor solutions with larger OF to be accepted, but in a limited way because the threshold value is fixed and always decreasing, with a very rigid control on the size of the OF difference, ΔOF. Simulated annealing uses a more flexible control on the values of the threshold, allowing transitions out of a local minimum at nonzero temperatures.

SA was first introduced by Kirkpatrick *et al.*, 1983) as an algorithm to solve very well known combinatorial optimization problems, reducing the risk of falling into local minima (or metastable solutions) common to iterative improvement methods. These authors proposed the use of the Metropolis (Metropolis *et al.*, 1953) procedure from statistical mechanics. This procedure generalizes iterative improvement by incorporating controlled uphill steps (to worse solutions). The procedure states the following: consider that the change in the objective function is ΔOF; if $\Delta OF \leq 0$, then the change in the system is accepted and the new configuration is used as the starting point in the next step; if $\Delta OF > 0$ then the probability that the change is accepted is determined by $P(\Delta OF) = exp(-\Delta OF / t)$ where t is a control parameter called temperature; a random number uniformly distributed in the interval (0,1) is taken and compared with the former probability; if this number is lower than $P(\Delta OF)$ then the change is accepted. The SA algorithm runs in the following way: i) the system is *melted* at a high

temperature (initial temperature, t_0); ii) the temperature is decreased gradually until the system *freezes* (because no better solutions are found and the probability of uphill steps is near zero); iii) at each iteration the Metropolis procedure is applied; iv) if any of the stopping criteria is reached the algorithm is stopped and the best solution found is presented.

The generic SA algorithm for a minimization, considering a neighborhood structure N, a solution space χ, and an objective function OF has the following pseudo-code.

```
Select an initial solution X_best;
Select an initial temperature t_1>0;
Select a temperature reduction factor;
Repeat
  Repeat
    Randomly select X ∈ N(X_best);
    ΔOF = OF(X) – OF(X_best);
      if ΔOF<0 then
                X_best = X
      else
              generate random z uniformly in (0,1);
      if z<exp(-ΔOF/t) then X_best = X;
  Until iterations = max_iterations
  Set t = α t;
Until stopping condition = true;
X_best is the optimal solution found.
```

In order to speed-up the process several improvements have been proposed, namely by limiting the number of iterations at each temperature, *i.e.*, defining the number *max_iterations*. The dimension of the Markov chain has been proposed to be a function of the dimension of the problem (Kirkpatrick *et al.*, 1983): temperature is maintained until 100Ω solutions (iterations), or 10Ω successful solutions have been tested, whichever comes first. Ω stands for the number of variables (stations) in a problem. These authors proposed also that the annealing is stopped (stopping criterion) if after three consecutive temperatures the number of acceptances is not achieved. It can also be considered that if the average value of the OF does not change after a pre-established number of temperature decreases (R_{STOP}), then the annealing is stopped. These parameters control the time spent at each temperature and the total running time. Along with these dynamic criteria, a static one may be used to halt the process when a minimum temperature, t_{min}, is reached. The former will guaranty that the annealing will stop if none of

the dynamic criteria is fulfilled, even before the total number of iterations is attained. In our algorithm both the dynamic and the static criteria were implemented.

The initial temperature, t_0, is calculated by running a fast (rapid temperature decrease) schedule and picking-up the temperature for which more than 95% of the iterations are accepted. Temperature is usually decreased with a constant rate, α, usually close to one (*e.g.,* 0.90 or higher). Aarts and Korst (1990) showed that SA can find optimal solutions if equilibrium is attained at each temperature (constant OF mean and variance) and proposed a temperature schedule dependent on OF variance that guaranties that. Despite this very attractive characteristic such schedule tends to converge too slowly. Other t schedules for optimality were also proposed by Geman and Geman (1984), Hajek (1988), and Siarry (1997). These however may not converge in a feasible amount of time for many problems (Cohn and Fielding, 1999). The wealth of practical experience with the faster t schedule used here indicate that the solutions found should be *good* local optimal ones. In practical terms: the local optimal solutions are a compromise between relatively good solutions in an amount of time significantly smaller than the one necessary to guaranty best quality solutions provided (in theory) by slower schedules.

The generation mechanism of a new solution, U, from the neighborhood, $N(X)$ of a solution X is a simple one: Consider the solution vector $X=\{x_1, \ldots, x_\omega\}$; i) randomly remove one element (station) from X; ii) randomly choose an element in $\chi \setminus X$; iii) insert this element in X, obtaining U; iv) confirm if U is a valid candidate solution (eq. 6.11), if yes continue, otherwise goto i).
A specific computer code in FORTRAN that incorporates both the mse and the SA algorithm was developed by the authors to optimize localization problems and adapted to this specific problem.

6.4 CASE STUDY

6.4.1 Study area and source data

The Sado Estuary is the second largest estuary in Portugal with an area of approximately 240 km^2. It is located on the west coast of Portugal, 45 km south of Lisbon. Most of the estuary is classified as a nature reserve. The Sado Estuary basin is subject to intensive land-use practices and plays an important role in the local and national economy. Most of the activities in the estuary (*e.g.*, industry, shipping, intensive farming, tourism and urban development) have negative effects on water and sediment physic-chemical quality and biotic communities (Caeiro *et al.*, 2002).

During the year 2001 sediment was sampled in 153 locations (stations) in an extensive estuarine sediment campaign (Caeiro *et al.*, 2002). Each sample was analyzed for fine fraction (FF), organic matter (OM) and redox potential (Eh). The data should contribute to define areas of similar physic-chemical characteristics and help designing a future sediment monitoring network. This will then be part of a Sado estuary management system.

6.4.2 Data processing

The definition of spatially homogenous physic-chemical areas is assumed as a necessary first methodological step (Caeiro *et al.*, 2002). These areas result from data collected in a sampling campaign, after some statistical transformations (Figure 6.1): i) principal component analysis (PCA) using data on log(FF), log(OM), and Eh; ii) variographic analysis on the first PCA component; iii) intervariable Euclidean distance (as opposed to geographic distance) computation; iv) computation of a dissimilarity matrix (Oliver and Webster, 1989) – equation (6.13); v) Euclidean distance computation on the dissimilarity matrix (cluster analysis); vi)

selection of four clusters and physic-chemical interpretation. A new variable, Z, is obtained by interpretation of cluster analysis of d_{ij}* projections, and represents stations that share common physic-chemical characteristics and similar spatial autocovariance.

$$d_{ij}^{*} = d_{ij} \times \frac{c}{c_0 + c} \times \left[1.5 \times \frac{|v_{ij}|}{r} - 0.5 \times \left(\frac{|v_{ij}|}{r} \right)^3 \right] + d_{ij} \times \frac{c_0}{c_0 + c} \quad \text{for } 0 < v_{ij} \leq r$$

$$d_{ij}^{*} = d_{ij} \text{ when } v_{ij} > r \qquad (6.13)$$

where: d_{ij} is the Euclidean data-space dissimilarity of log(OM), log(FF) and Eh between sample sites; c is the variogram sill; c_0 is the nugget variance; r is the range; v_{ij} are the Euclidean geographic distances between locations i and j. Mean values and confidence intervals (α=0.05) of Z_i, i= 1,...,4, are shown on Table 6.1 and reflect four different physic-chemical homogeneous areas found in the sampling campaign. More information on the identification of the homogeneous areas may be found in the recently published work by Caeiro et al. (2003).

Table 6.1. Physic-chemical sediment parameters of each homogeneous areas

	Cut-offs			
Parameter	z_1	z_2	z_3	z_4
% OM	8.6 ± 2.4	4.2 ± 1.4	1.9 ± 0.7	0.9 ± 0.3
% FF	60.4 ± 27	21.7 ± 11.8	9.1 ± 7.8	1.5 ± 1.3
Eh	-278.9 ± 68.6	-178.8 ± 72.6	-137.4 ± 50.9	74.4 ± 49

It is then intended to estimate the probability distribution function of Z considering four cut-offs: 1, 2, 3, and 4. The indicator transform is given by equation (6.1) with cut-off z_c = 1,…4. For easiness of identification indicators have a subscript equal to the cut-off value.

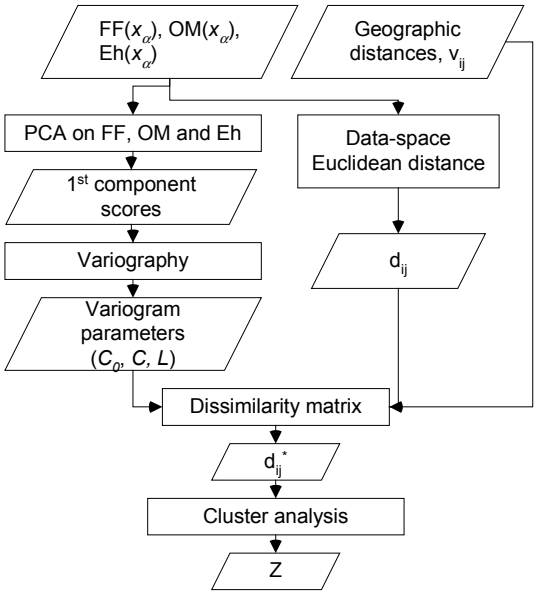


Figure 6.1. Flowchart for the definition of the sediment homogenous areas

Indicators frequencies, $\phi(A,z_c)$, and the proportion of stations with values in the interval $]z_{c-1}, z_c]$, $\psi(A,z_c)$, are shown in Table 6.2. As Z can only take integer values, the proportions correspond to the stations for which $z(x_\alpha) \leq z_c$.

Table 6.2. Indicators frequencies, $\phi(A,z_c)$ and $\Psi(A,z_c)$

	Indicators			
	i_1	i_2	i_3	i_4
ϕ	0.1176	0.4967	0.7320	1
ψ	0.1176	0.3791	0.2353	0.2680

The maximum number of stations to be included in the sediment monitoring network had necessarily to be less than 100 for budgetary reasons, but if possible a much lower value should be looked for (ω_{Max}). Moreover the new design must reflect the sediment physic-chemical variability detected with the prior sampling campaign, and presented in the previous section. Therefore it is asked that the proportions of monitoring stations in each of the

identified homogeneous areas are similar to the proportions in the sampling campaign. This amounts to make the candidate solutions to have $\Psi_{S'}(A,z_c) \approx \Psi_S(A,z_c)$.

Three different conditionings on the objective function are studied: i) no conditioning on the proportions is imposed; ii) conditioning with $\delta = 0.5$; iii) conditioning with $\delta = 0.3$. In the first the entire solution space is a feasible space, while in the others a solution is only feasible if it respects the condition. More stringent conditions were also tested ($\delta < 0.3$), but resulted in too long processing times and no solutions were obtained. It will be seen bellow that such conditioning may not be necessary.

For each OF conditioning several network dimensions were tested according to the following scheme: i) impose a maximum number of monitoring stations (ω_{Max}) to be included in the new design; ii) find the optimal allocation solution with SA; iii) increase ω_{Max} and return to i).

Eight different monitoring network dimensions (cardinality of S': ω_{Max}) are tested, {30,40,50,60,70,80,90,100}. SA solutions are considered optimal when more than 70% out of 20 consecutive runs with the same objective function conditions (ω_{Max}, δ) and SA parameters have the lowest and equal s^2 value. Runs were made on PC Intel 2000 MHz machines.

6.5 OPTIMIZATION RESULTS AND DISCUSSION

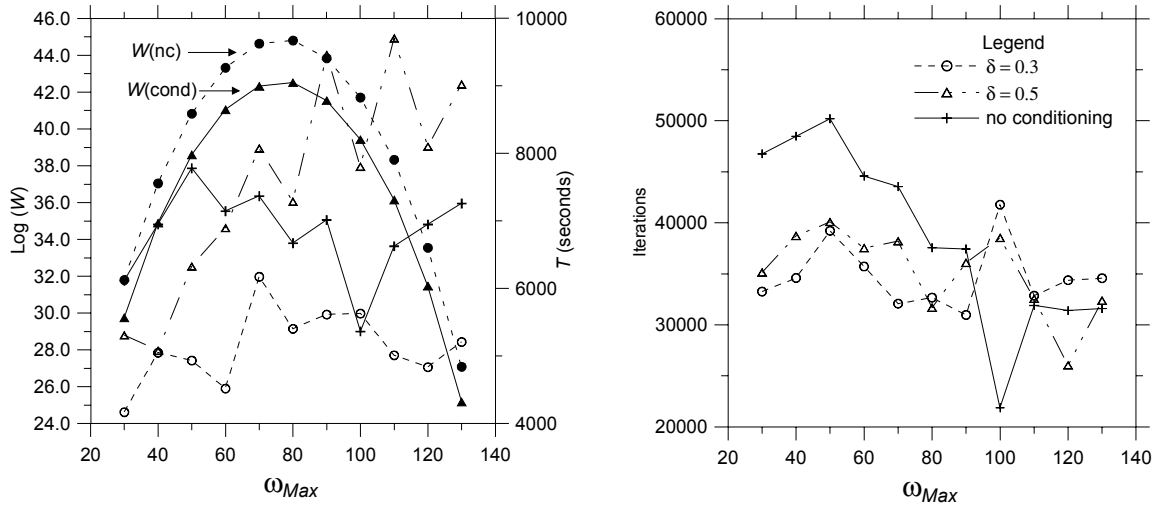
6.5.1 Feasible space

The number of combinations of Ω sampling stations with ω possible monitoring stations is given by the well-known formula $W = \Omega!/((\Omega-\omega)!\omega!)$. Now, if one wants to calculate the combinations conditioned to the reproduction of the proportions the expression becomes

$$W = \prod_{i=1}^{k} \frac{\Omega_i!}{(\Omega_i - \omega_i)!\,\omega_i!} \qquad (6.14)$$

where i is the indicator number, Ω_i the number of sampling stations with indicator i, and ω_i the number of monitoring stations with indicator i imposed by conditioning. The number of combinations in each case is represented in Figure 6.2a, in logarithmic scale, for different ω numbers. Conditioning reduces the dimension of the feasible space by more than two orders of magnitude when $\omega_{Max} \approx 77$ (from $7.3\text{x}10^{44}$ to $3.6\text{x}10^{42}$). The lowest W value is still higher than 10^{25}, indicating that any attempt to solve even the lowest dimension combinatorial problem exhaustively would take (in the same machine) more than 10^6 times the age of the universe! Despite this practical difficulty it is interesting to see how the optimization problem structure (intrinsic to each specific problem) affects the time necessary to find an optimal solution (total running time), T, and the number of iterations. The time is dependent on the number of iterations and on the time necessary to compute the objective function. The number of iterations is however strongly dependent on the structure of the problem and not necessarily on the dimension of the feasible space, otherwise this parameter would have followed more closely W. In reality it is observed that the problem structure changed for each ω_{Max} value (Figure 6.2b). However, when no conditioning is imposed the number of iterations decreases up to $\omega_{Max} = 100$ (Figure 6.2b and Table 6.3), stabilizing after that. This may indicate that a higher proportion of indicators one and two, which had always higher estimation errors (yet unpublished results), contribute to a more structured problem, possibly with less local minima. Such behavior is not clear when conditioning is imposed due to a forced proportion of these indicators in all tested ω_{Max} values. As a consequence T varies around a constant mean for no conditioning (as ω increases the number of iterations decreases and the OF computing time increases) (Table 6.2b); when conditioning is imposed T increases up to $\omega_{Max} = 90$, with tendency to stabilize after that (therefore dependent essentially on the OF computing time).

It is also interesting to see that strong conditioning (δ = 0.3) results in the lowest T, while weaker conditioning (δ = 0.5) results in the highest T. Once again the problem structure plays a fundamental part. This may not, however, be extrapolated to other problems because it is problem dependent.



a) b)

Figure 6.2. Effect of conditioning and cardinality of S': a) Dimension of the solution space (W) and time until optimal solution (T); b) Number of iterations. *nc* and *cond* in $W()$ stand for without and with conditioning, respectively. Legend on figure a) the same as in b)

Table 6.3. Results for different ω_{Max} and δ values: mean square estimation error (s^2), number of iterations (Iter.) and time necessary to reach an optimal solution (T) in seconds

	No conditioning			δ = 0.5			δ = 0.3		
ω_{Max}	s^2	Iter.	T(s)	s^2	Iter.	T(s)	s^2	Iter.	T(s)
30	0.12889	46760	6114	0.62667	35135	5306	1.00000	33260	4165
40	0.12000	48484	6946	0.62500	38729	5076	0.81938	34596	5046
50	0.11840	50209	7778	0.55000	40088	6325	0.68000	39215	4930
60	0.13639	44582	7142	0.54000	37523	6894	0.58750	35718	4515
70	0.18694	43560	7366	0.41429	38201	8071	0.58551	32078	6171
80	0.24938	37568	6669	0.41817	31705	7284	0.57000	32668	5400
90	0.33099	37441	7012	0.41421	36087	9456	0.56000	30972	5612
100	0.50910	21887	5361	0.41613	38523	7803	0.55700	41776	5626

6.5.2 Optimal conditioning

Conditioning on the objective function has an interesting effect on the errors of the estimated indicators proportions. When no condition is imposed the error in the estimated proportions (($\Psi_S(A,z_c)$- $\Psi_{S'}^*(A,z_c)$) is higher than 0.3 for low ω_{Max} values and decreases with growing ω_{Max} (Figure 6.3a). Imposing the conditioning on the proportions causes a reduction in the error of about 50 % for δ = 0.5 and about 25 % for δ = 0.3 (Figure 6.3b and c) when compared with no conditioning. Moreover, (Ψ_S (A,z_c) - $\Psi_{S'}^*(A,z_c)$) are one third of that of the imposed interval (Ψ_S (A,z_c) - $\Psi_{S'}(A,z_c)$), for both δ = 0.5 and 0.3. This may indicate that imposing lower δ values would lead to similar results. If this is true then imposing δ< 0.3 would also lead to very narrow intervals and eventually to a very limited number of neighboring feasible solutions. Such behavior is concordant with the aforementioned practical difficulty – extremely long processing times. Furthermore, with δ = 0.3, $\Psi_{S'}^*(A,z_c)$ estimation error is lower than 10 %, which is considered an acceptable error.

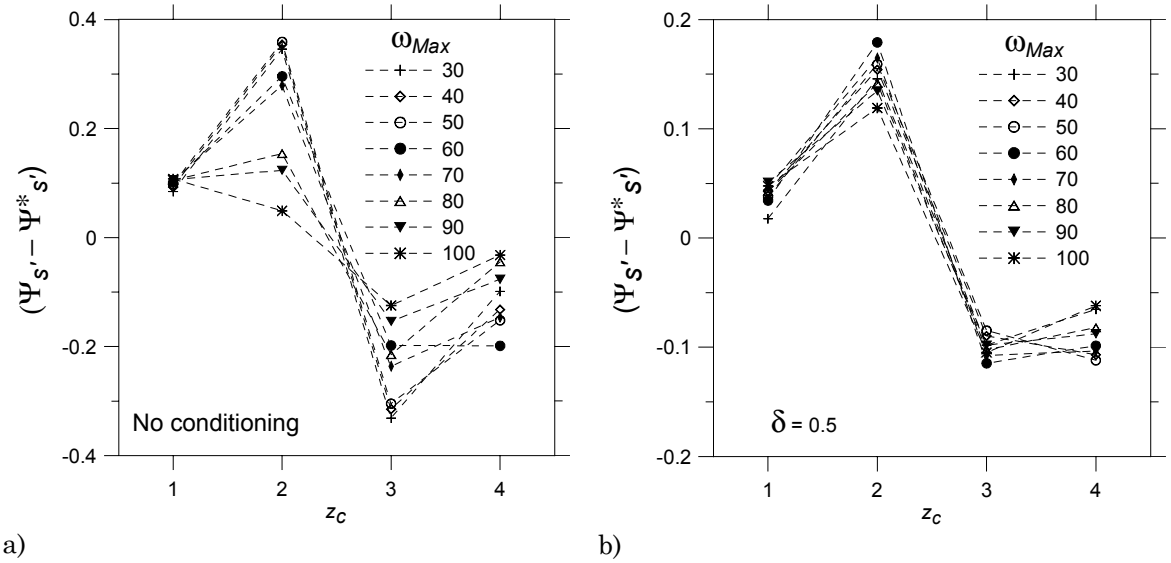
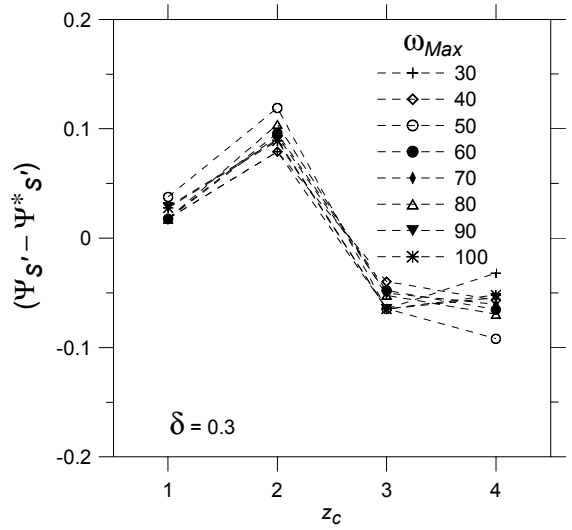


Figure 6.4. Relative error when estimating $\Psi_{S'}(x_\alpha,z_c)$: a) no conditioning; b) δ= 0.5; c) δ=0.3



c)

Figure 6.3. Relative error when estimating $\Psi_{S'}(x_\omega z_c)$: a) no conditioning; b) $\delta = 0.5$; c) $\delta=0.3$. (cont)

When high or low values of a variable are clustered in small areas scattered in the study area, their relative frequencies are low or data is too random, variogram fitting becomes difficult and prone to error. The result is the fitting of theoretical variograms that only roughly approximate the real variability. This does not hinder the geostatistical method, but justifies the need to consider the conditioning on the proportions. An example of such need is shown by indicators one and two, for which the estimation errors are higher, therefore leading the optimization algorithm to select preferentially the two remaining indicators with lower estimation errors. As a result, the latter have in all the studied cases proportions higher than in the original data set, as a way to compensate the bias introduced with the first two indicators. However, conditioning significantly reduces the bias. Another, even more important effect of estimation errors, is reflected in Figure 6.4: if no conditioning is used increasing the number of stations will result in higher estimation error. This is opposite to what is expected because when increasing the number of points available for estimation the accuracy of the estimated value should decrease. This inversion may be explained considering that with very low ω only stations with low estimation error are included in the optimal solution; as ω

increases more higher estimation error stations are included. Clearly if no conditioning is imposed the monitoring network is dominated by the last two indicators (Figure 6.5a).

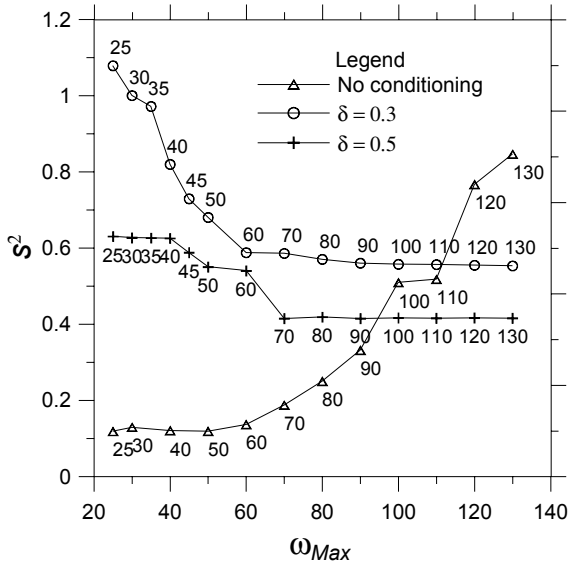
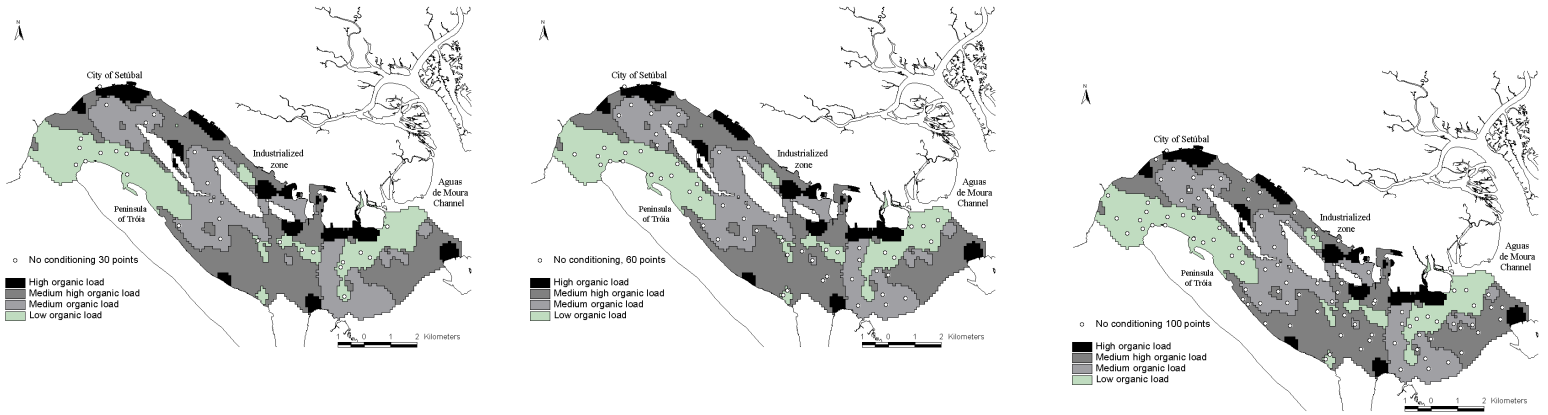


Figure 6.4. Mean square estimation error, s^2, without conditioning and with conditioning (δ = 0.3; δ=0.5)

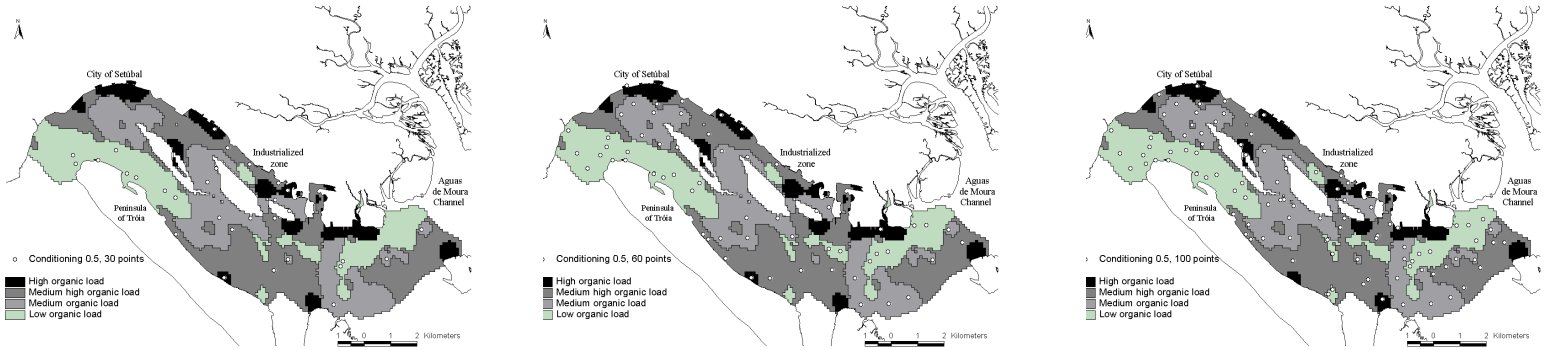
When conditioning is used the expected increase in accuracy is observed (Figure 6.4), and the stronger the conditioning the higher the s^2 because more high-error stations are imposed at lower ω_{Max} values.

Figure 6.5b and c show the resulting monitoring networks with different ω_{Max} values. The proportions of the first two indicators are higher in these cases, and with δ = 0.3 a better reproduction of the probabilities is obtained, considered as an important decision criterion to monitor the homogeneous areas.



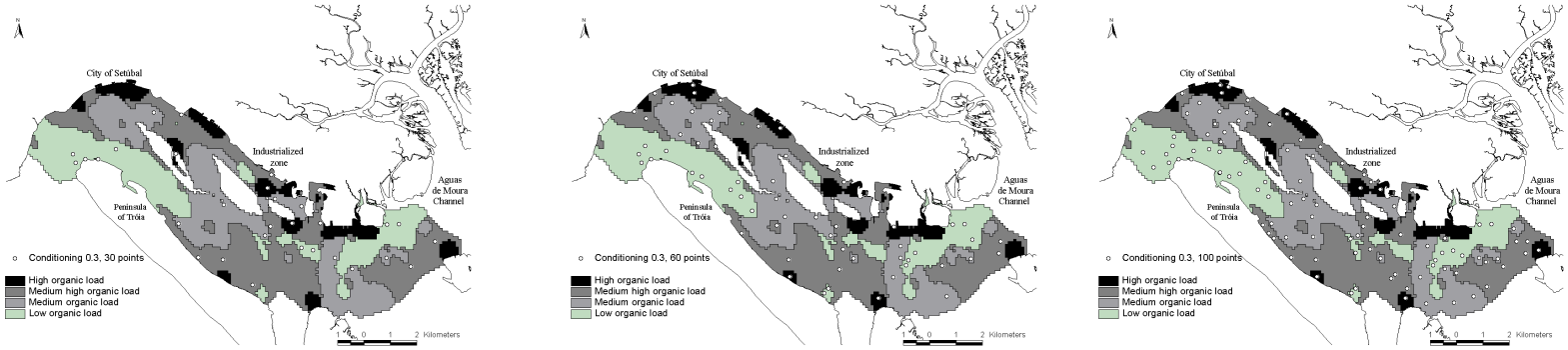
a1) 30 stations a2) 60 stations a3) 100 stations

Figure 6.5. Monitoring networks for different ω_{Max} values: a) no conditioning (nc)



b1) 30 stations b2) 60 stations b3) 100 stations

Figure 6.5. Monitoring networks for different ω_{Max} values:; b) $\delta = 0.5$



c1) 30 stations

c2) 60 stations

c3: 100 stations

Figure 6.5. Monitoring networks for different ω_{Max} values: c) δ=0.3

6.5.3 Optimal monitoring network

A monitoring network dimension is considered optimal if after some ω value each new added station had little effect on the monitoring spatial accuracy, s^2, *i.e.*, if the marginal gains are small. The gains are shown in Figure 6.4. Gains in accuracy are high up to the 60th station, becoming much less important after that. Sixty is therefore considered as the optimal ω value. The resulting network is shown in Figure 6.5. Had no conditioning or $\delta = 0.5$ been chosen and the optimal number of stations would be similar, though the spatial distribution of stations quite different (*cf*. Figure 6.4 and Figure 6.5).

6.6 CONCLUSIONS

The following conclusions are drawn: i) Objective function conditioning is necessary to guaranty reproduction of indicator variables probability density functions; ii) the highest the conditioning the closest the posterior (estimated) distribution function is to the prior (data) distribution function; iii) conditionings with δ<0.3 originates extremely long running times and was shown to be unnecessary; iv) if no conditioning is used, s^2 with increasing number of monitoring stations as a result of the bias introduced by variogram fitting errors; v) the time necessary for SA to reach a solution is, in this particular case-study, more dependent on the structure of the problem than on its dimension; vi) the optimization problem studied here can not be solved exhaustively due to the enormous number of possible combinations that would have to be tested; vi) the solution attained may not be the global optimal one, but should be a good quality one – a solution very close to the global minimum attained in a feasible amount of time; vii) a sediment monitoring network with sixty stations was obtained. By construction

this network has a proportion of station inside each sediment homogeneous areas similar to the proportions in the prior sampling program with 153 sampling stations.

7 WORK CONCLUSIONS

In this section the most relevant conclusions obtained in the former sections are summarized and discussed together.

The third and fourth sections propose the optimization of groundwater monitoring networks with complete data series in each measuring station. Different objective function models were tested by incorporating: (1) a term related to the accuracy (or reliability) of the spatial distribution of stations, in the form of the variance of estimation error obtained by kriging; (2) a term related to the amount of redundancy between stations, in the form of the relative difference between their time series, (3) a term related to the time necessary to measure at each station, and (4) a term related to the time necessary to travel along the monitoring network. The last two terms are a measure of the financial cost associated with the network exploitation.

Four variants on the objective function were tested: (1) with only the first term, (2) with only the second term, (3) with the first two terms, (4) with all terms. Results showed that in the first case the optimal distribution of stations is an even one. If the kriging variance of the

estimation error (kriging variance) is used only the variogram model and the geometric configuration of stations influence the objective function values. A more even distribution is by construction the configuration with lowest variance. However, if the variance of the estimation error is calculated using both the true values and the estimated ones (fictitious point variance), the spatial distribution is more clustered because accuracy considers, in this case, the value taken at each location. This is shown in the fifth section. A true comparison between these approaches would require testing both with the same conditions, but that was not set as an objective for this work. Some remarks are possible, though. Kriging variance may be a good option when used for network augmentation because values at unsampled locations are unknown, and for network reduction when only one variogram model is used and considered representative of spatial correlation for all measuring times.

When only data redundancy is considered the spatial distribution reflects the difference in local state variable responses to external signals after being filtered by the supporting system. It is therefore a measure different from spatial accuracy where averaging is made over the entire field. If two stations, though near one another, have quite different time series they will have higher probability of being included in the optimal design than two stations far apart but with similar time series. Thus, some clustering may appear, as was shown in the third section, near zones of high variability. An objective function with only a temporal redundancy term is necessarily incomplete, as it is one with only a spatial accuracy term.

Including both measures in the objective function enhances the overall quality of the resulting network. In this case not only the spatial autocorrelation structure is considered but also the local temporal variability. The optimal spatial distribution of monitoring stations tends to be more clustered than when only kriging variance is used, but less than when using only temporal redundancy.

Now if exploitation costs are included in the objective function, the resulting monitoring network is optimized such that data representativity is a part of the monitoring system accounted together with the economical component. This approach is not equal to constrain the statistical component objective function to a fixed budget. Here exploitation costs are optimized also. In the particular case studied in the fourth section a diminution in monitoring times was not accompanied by a diminution in travel times. The latter depends on the paths available between stations, hence a particular set of stations with optimal data representativity and monitoring times may correspond to a longer travel path (some connections between stations are not available). This seems to happen in the studied example. Results from the fourth section also showed that it is possible to evaluate the optimal monitoring network dimension with this objective function by balancing the gains in the information collected with the network and its exploitation costs.

Exploitation costs minimization is also discussed in the fifth section, but there, unlike the former sections, data from a real case-study is very incomplete. The method proposed to handle sparse data matrixes is based in some expertise judgment as to which measuring times should be used for network optimization. A selection based in factorial analysis is proposed and discussed. Other statistical or empirical criteria might have been used. The method proposed should be robust enough to generate optimal monitoring networks capable of detecting the main state variable variability, as long as the measuring times selected for optimization are representative of the phenomenon under study. If seasonality or tendencies are detected in temporal data, measuring times should be carefully selected to reproduce them. Results showed that in the example case-study, where nitrate concentrations showed seasonality, principal component analysis may be a good tool to help selecting the most representative measuring times. Important exploitation costs reductions may be achieved with optimization. In the Gabbros de Beja case a reduction in costs of more than 80% was attained by compromising less than 10% in the mean precision, and 25% in local precision. The method

showed also to be robust as it gave good interpolation and extrapolation results. It may become a helpful tool to optimize European networks for agricultural nitrate contamination detection.

The sixth section presents monitoring network optimization example usually handled by empirical stratification, i.e., once identified different homogeneous areas these are considered separately. The propose method allows to optimize strata simultaneously by conditioning the objective function in an innovative way. Results showed that the necessary stratification is also obtained in the optimal networks. Reductions in exploitation costs of more than 60% were attained in the Sado estuary sediment monitoring network example.

The methods presented in this work share a common feature: though demonstrated with specific practical problems, they are generally applicable to environment monitoring networks. This generality was exemplified in the third and fourth sections where no environmental state variable was identified. It is not possible to include in one single work all the possible practical problems found when reducing the dimension of an existing network, nor was it the objective. However, the most common general types of problems were included, and we think simulated annealing has shown to be a flexible enough algorithm to solve such problems. Combinations of the proposed methods are easily made to accommodate other particular conditions.

Some important factors also important when altering an existing monitoring network, not discussed thus far, and which may be considered in future are:

1. The incorporation/quantification of measurement errors;
2. to include more than one state variable;
3. the incorporation of data accuracy (but difficult to quantify as measurement error);
4. to include soft information;
5. to consider fuzzy approaches to both soft data and variogram parameters;

6. the incorporation of sampling frequency in the analysis.

Measurement errors are easily incorporated in the objective function, be it in the spatial domain (incorporated in the kriging variance via the nugget effect in the variogram models), or in the time domain (*e.g.*, with penalty functions).

Considering more than one state variable may improve spatial accuracy by considering also cross-covariances in the kriging estimations.

Data accuracy refers to the measurement of its quality in situations in which field data collection or laboratory analysis procedures fail to produce high-quality data. Typical problems (or doubts) with data accuracy arise when well construction is not adequately documented; field data collection procedures are inappropriate; sample collection procedures are inappropriate; sample analysis procedures are inadequate or are undocumented; well screens are inappropriately placed to monitor the intended water quality change; well was poorly constructed or has suffered degradation (causing, *e.g.*, mixing of waters, or altering water quality by chemical interaction); wells may not be sampling the intended stratigraphic unit; samplers may be pushed away from the prescribed location by water currents (in the case of estuarine or river networks); etc. Data representativity is not easy to quantify and may again require the use of fuzzy-type intervals to empirically defined quantities.

The incorporation of soft information may be useful if one wants to: i) introduce low quality information in the estimation functions (*e.g.*, geophysical data); or ii) introduce empirically determined weighting factors, that may be related with data representativity or with some other quantity not directly related with the measured variables (*e.g.*, proximity to the offices, to some restaurant, person responsible for making the measurements).

Fuzzy variogram parameters is a way of incorporating uncertainty in the data, by transferring it into the variogram model in a nonfrequentist approach, or a way to construct variogram models when no information is available (in this case the variogram model is constructed by empirical extrapolation obtained from experience or from similar cases). Bardossy *et al.* (1988) and Bardossy *et al.* (1990a) set the theoretical basis of kriging with imprecise variograms and Bardossy *et al.* (1990b) showed one practical application. Some unpublished results made with our data and variogram models showed that unless the range of the membership functions was very wide, the resulting kriging variance (also a fuzzy number) would not be significantly altered. However, this approach may be useful if variogram model parameters are not available. Bogardi *et al.* (1985) present interesting results of fuzzy kriging for new network optimization and conclude that with this approach the most cost-effective network may not be a regular one, as would be obtained if non-fuzzy kriging variance had been used.

In what regards sampling frequency, though not handled here in a direct way, it was discussed in the introduction that in many cases (the ones considered here) frequency is imposed a priori. Nevertheless, it would be interesting to evaluate the effect of sampling frequency in the resulting monitoring networks.

The previous list is not exhaustive and points-out only some of the developments related directly with the methods. Other natural developments include the application of the methods to different environment compartments.

REFERENCES

Aarts, E. and J. Korst, *Simulated annealing and Boltzman machines*, John Willey and Sons, New York, 1990.

Aarts, E. and P. J. M. Laarhoven, Statistical cooling: a general approach to combinatorial optimization problems, *Philips Journal of Research,* **40**(4), 193-226, 1985.

Ahlfeld, D. P. & Pinder, G. F.*, A ground water monitoring network design algorithm* ,edited by Ahlfeld D P and Pinder G F, Department of Civil Engineering and Operations Research, Princeton University, Report 87-WR-4, Princeton, N.J., 1988.

Ahmed, S., de Marsilly, G. & Talbot, A., Combined use of hydraulic and electrical properties of an aquifer in a geostatistical estimation of transmissivity, *Ground Water,* **26** (1), 78-85, 1988

Alli, M. M., E. A. Nowatzki and D. E. Myers, Probabilistic analysis of collapsing soil by indicator kriging, *Mathematical Geology,* **22**(1), 15-38, 1990.

Amorocho, J. and B. Espildora, Entropy in the assessment of uncertainty of hydrologic systems and models, *Water Resources Research,* **9**(6), 1511-1522, 1973.

Anderssen R. S. Global optimization, in *Optimization,* edited by R. S. Anderssen, L. S. Jennings and D. M. Ryan, pp. 1-15, University of Queensland Press, Queensland, Australia, 1972.

Angulo, M. and W. H. Tang, Optimal ground-water detection monitoring system design under uncertainty, *Journal of Geotechnical and Geoenvironmental Engineering,* **125**(6), 510-517, 1999.

ASCE, Review of geostatistics in geohydrology. I. Basic concepts, *Journal of Hydraulic Engineering,* **116**(5), 612-632, 1990a.

ASCE, Review of geostatistics in geohydrology. II. Applications, *Journal of Hydraulic Engineering,* **116**(5), 633-658, 1990b.

Bardossy, A., I. Bogardi and W. E. Kelly, Imprecise (Fuzzy) information in geostatistics, *Mathematical Geology,* **20**(4), 287-311, 1988.

Bardossy, A., I. Bogardi and W. E. Kelly, Kriging with imprecise (Fuzzy) varigrams; I: theory, *Mathematical Geology,* **22**(1), 63-79, 1990a.

Bardossy, A., I. Bogardi and W. E. Kelly, Kriging with imprecise (Fuzzy) varigrams; II application, *Mathematical Geology,* **22**(1), 81-94, 1990b.

Becker, R. W. and G. V. Lago, A global optimization algorithm, paper presented at *Proceedings*

of the 8th Allerton Conference on Circuits and Theory, 1970.

Berbee, H. C. P., C. G. E. Boender, A. H. G. Rinnooy Kan, R. L. Scheffer, R. L. Smith and J. Telgen, Hit-and-run algorithms for the identification of nonredundant linear inequalities, *Mathematical Programming,* **37**(2), 184-207, 1987.

Berkowitz, B. and D. P. Hansen, A numerical study of the distribution of water in partially saturated porous rock, *Transport in Porous Media ,* **45**(2), 303-319 , 2001.

Boender, C. G. E., R. J. Caron, J. F. McDonald, A. G. H. Rinnooy Kan, H. E. Romeijn, R. L. Smith, J. Telgen and A. C. F. Vorst, Shake.and.bake algorithms for generating uniform points on the boundary of bounded polyhedra, *Operations Research,* **39**(6), 945-954, 1991.

Boender, C. G. E., A. H. G. Rinnooy Kan, L. Stougie and G. T. Timmer, A stochastic method for global optimization, *Mathematical Programming,* **22**(2), 125-140, 1982.

Boender C. G. E.andH. E. Romeijn , Stochastic methods, in *Handbook of Global Optimization,* vol Volume 2, edited by R. Horst and P. M. Pardalos, pp. 829-869, Kluwer Academic Publishers, Netherlands, 1995.

Bogardi, I., A. Bardossy and L. Duckstein, Multicriterion network design using geostatistics, *Water Resources Research,* **21**(2), 199-208, 1985.

Bras, R. L. and I. Rodríguez-Iturbe, Network design for the estimation of areal mean of rainfall events, *Water Resources Research,* **12**(6), 1185-1195, 1976.

Bras, R. L. & Rodríguez-Iturbe, I., *Rainfall-runoff as spatial stochastic processes: data*

collection and synthesis, Ralph M. Parsons Lab. for Water Resources and Hydrodynamics, Dep. of Civil Eng., Mass. Inst. of Technology, Tech. Rep. 196, Cambridge, U.S.A., 1975.

Brooks, S. H., A discussion of random methods for seeking maxima, *Operations Research,* **6**(2), 244-251, 1958.

Brus, D. J., M. J. W. Jansen and W. F. de Haan, Designing efficient sampling schemes for reconnaissance surveys of contaminated bed sediments in water courses, *Geologie en Mijnbouw-Netherlands Journal of Geosciences ,* **79**(4), 441-447 , 2000.

Buxton B. E.andA. D. Pate , Joint temporal-spatial modeling of concentrations of hazardous pollutants in urban air, in *Geostatistics for the Next Century,* edited by R. Dimitrakopoulos, pp. 75-87, Kluwer Academic Publishers, Netherlands, 1994.

Bélisle, C. J. P., Convergence theorems for a class of simulated annealing algorithms on $\Re^d$, *Journal of Applied Probability,* **29**(4), 885-895, 1992.

Bélisle, C. J. P., H. E. Romeijn and R. L. Smith, Hit-and-run algorithms for generating multivariate distributions, *Mathematics of Operations Research,* **18**(2), 255-266, 1993.

Caeiro, S., P. Goovaerts, M. Painho, M. H. Costa and S. Sousa, Optimal spatial sampling design for mapping estuarine sediment management areas, paper presented at *5th AGILE Conference on Geographic Information Science,* Universitat de les Illes Balears, Spain, 2002.

Caeiro, S., P. Goovaerts, M. Painho and M. H. Costa, Delineation of estuarine management areas using multivariate geostatistics: the case of Sado estuary, *Environmental Science and Technology*, **37**, 4052-4059.

Candela, L., R. A. Olea and E. Custodio, Lognormal kriging for the assessment of reliability in groundwater quality control observation networks, *Journal of Hydrology,* **103**, 67-84, 1988.

Carpaneto, G., M. Dell'Amico and P. Toth, Algorithm 750: CDT: a subroutine for the exact solution of large-scale, asymetric traveling salesman problems, *ACM Transactions on Mathematical Software,* **21**(4), 410-415, 1995a.

Carpaneto, G., M. Dell'Amico and P. Toth, Exact solution of large-scale, asymetric traveling salesman problems, *ACM Transactions on Mathematical Software,* **21**(4), 394-409, 1995b.

Caselton, W. F. and T. Husain, Hydrologic networks: information transmission, *Journal of the Water Resources Planning and Management Division,* **106**(WR2), 503-519, 1980.

Chandler, B., C. Rekeczky, Y. Nishio and A. Ushida, Adaptive simulated annealing in CNN template learning, *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences,* **E82A**(2), 398-402, 1999.

Chen, S., R. Istepanian and B. L. Luk, Digital IIR filter design using adaptive simulated annealing, *Digital Signal Processing,* **11**(3), 241-251, 2001.

Cochran, W. G., *Sampling Techniques*, John Wiley & Sons, Inc., New York, 1977.

Cohn, H. and M. Fielding, Simulated annealing: searching for an optimal temperature schedule, *SIAM Journal on Optimization,* **9**(3), 779-802, 1999.

Collins, N. E., R. W. Eglese and B. L. Golden, Simulated annealing - an annotated bibliography, *American Journal of Mathematical and Management Sciences,* **8**(3/4), 209-307,

1988.

Connolly, D. T., An improved annealing scheme for the QAP, *European Journal of Operations Research,* **46**, 93-100, 1988.

Cressie, N., *Statistics for Spatial Data*, Wiley, N.Y, 1991.

Cunha, M. C. and J. Sousa, Water distribution network design optimization: simulated annealing approach, *Journal of Water Resources Planning and Management,* **125**(4), 215-221, 1999.

Cunha, M. D., On solving aquifer management problems with simulated annealing algorithms, *Water Resources Management,* **13**(3), 153-169, 1999.

David, M., *Geoestatistical Ore Reserve Estimation*, Elsevier Pub, Amsterdam, The Netherlands, 1977.

Dekkers, A. and E. Aarts, Global optimization and simulated annealing, *Mathematical Programming,* **50**(3), 367-393, 1991.

Delhomme, J. P., Kriging in the hydrosciences, *Advances in Water Resources,* **1**(5), 251-266, 1978.

Delhomme, J. P. and P. Delfiner, Application du krigeage à l'optimisation d'une campagne pluviométrique en zone aride, paper presented at *Syposium on the Design of Water Resources Projects With Inadequate Data,* Madrid, 1973.

Deutsch, C. V. and A. G. Journel, *GSLIB, Geostatistical software library and user's guide*, Oxford University Press, New York, USA, 1992.

Deyer, M. E. and A. M. Frieze, Computing the volume of convex bodies: a case where randomness provably helps, paper presented at *PSAM: Proceedings of the 44th Symposium in Applied Mathematics, Probabilistic Combinatorics and Its Applications,* 1992.

Dimitrakopoulos R.and X. Luo, Spatiotemporal modelling: covariances and ordinary kriging systems, in *Geostatistics for the Next Century,* edited by R. Dimitrakopoulos, pp. 88-93, Kluwer Academic Publishers, Netherlands, 1994.

Dixon, L. C. W.*, Global optima without convexity*, edited by Dixon L C W, Hatfield Polytechnic Hatfield, England, 1978.

Dixon, W. and B. Chiswell, Review of aquatic monitoring program design, *Water Resources,* **30**(9), 1935-1948, 1996.

Dixon, W., G. K. Smyth and B. Chiswell, Optimized selection of river sampling sites, *Water Resources,* **33**(4), 971-978, 1999.

Dorigo, M., V. Maniezzo and A. Colorni, The ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics - Part B,* **26**(1), 1-13, 1996.

Dougherty, D. E. and R. A. Marryott, Optimal groundwater-management .1. Simulated annealing, *Water Resources Research ,* **27**(10), 2493-2508 , 1991.

Duque, J. M., *Caracterização hidrogeológica e modelação matemática do aquífero dos Gabros de Beja*, MSc Thesis, University of Lisbon, FC, Lisbon, 1997.

Durand, M. D. and S. R. White, Trading accuracy for speed in parallel simulated annealing with simultaneous moves, *Parallel Computing,* **26**(1), 135-150, 2000.

EEA, Environment in the European Union at the turn of the century, Environmental assessment report No 2, edited by European Environment Agency, Copenhagen , 1999.

EEA, Environmental signals 2002 - Benchmarking the millennium. Environmental assessment report No 9, edited by European Environment Agency, Copenhagen, 2002.

EU, Council Directive 80/68/EEC of 17 December 1979 on the protection of groundwater against pollution caused by certain dangerous substances, *Official Journal of the European Union ,* **L**(20), 43-48, 1980.

EU, Council Directive 91/676/EEC of 12 December 1991 concerning the protection of waters against pollution caused by nitrates from agricultural sources, *Official Journal of the European Union,* **L**(375), 1-8, 1991.

EU, Council Directive 98/83/EC of 3 November 1998 on the quality of water intended for human consumption, *Official Journal of the European Union ,* **L**(330), 32-54, 1998.

Fallat, M. R. and S. E. Dosso, Geoacoustic inversion for the Workshop '97 benchmark test cases using simulated annealing, *Journal of Computational Acoustics ,* **6**(1-2), 29-43 , 1998.

Ferreyra, R. A., H. P. Apezteguia, R. Sereno and J. W. Jones, Reduction of soil water spatial

sampling density using scaled semivariograms and simulated annealing, *GEODERMA,* **1190**(3-4), 265-289 , 2002.

Fielding, M., Simulated annealing with an optimal fixed temperature, *SIAM Journal on Optimization,* **11**(2), 289-307, 2000.

Fonseca, P., *Estudo da Sutura Varisca no SW Ibérico nas regiões de Serpa-Beja-Torrão, Alvito-Viana do Alentejo*, PhD Thesis, Universidade de Lisboa, Lisboa, 1995.

Freeze, R. A., A Stochastic-conceptual Analysis of One-dimensional Groundwater Flow in Nonuniform Homogeneous Media, *Water Resources Research,* **11**(5), 725-741, 1975.

Freeze, R. A. and S. M. Gorelick, Convergence of stochastic optimization and decision analysis in the engineering design of aquifer remediation, *Ground Water,* **27**(6), 934-954, 1999.

Geman, S. and D. Geman, Stochastic Relaxation, Gibbs Distributions and the Baysian Restoration of Images, *IEEE Transaction On Patern Analysis And Machinee Inteligence,* **PAMI-6**(6), 721-741, 1984.

Glover R. E. The pattern of fresh-water flow in a coastal aquifer, in *Sea Water in Coastal Aquifers,* edited by U.S. Geological Survey Water-Supply Paper 1613., C32-C35, U.S. Geoogical Survey, 1964.

Grabow, G. L., C. R. Mote, W. L. Sanders, J. L. Smoot and D. C. Yoder, Groundwater monitoring network using minimum well density, *Water Science and Technology,* **28**(3-5), 327-335, 1993.

Hajek, B., Cooling schedules for optimal annealing, *Mathematics of Operations Research,* **13**(2), 311-329, 1988.

Harmancioglu, N. and N. Alspaslan, Water quality monitoring network design: a problem of multi-objective decision making, *Water Resources Bulletin,* **28**(1), 179-192 , 1992.

Harmancioglu, N. and V. Yevjevich, Transfer of hydrologic information among river points, *Journal of Hydrology,* **91**(1/2), 103-118, 1987.

Harmancioglu N. B., M. N. Alpaslan, and V. P. Singh , Needs for environmental data management, in *Environmental Data Management,* edited by N. B. Harmancioglu, M. N. Alpaslan and V. P. Singh, pp. 1-12, Kluwer Academic Publishers, Dordrecht, 1998.

Harmancioglu, N. B., O. Fistikoglu, S. D. Ozkul, V. P. Singh and M. N. Alspaslan, *Water Quality Monitoring Network Design*, Kluwer Academic Publishers, The Netherlands, 1999.

Hoeksema, R. J. and P. K. Kitadinis, Analysis of the Spatial Structure of Properties of Selected Aquifers, *Water Resources Research,* **21**(4), 563-572, 1985a.

Hoeksema, R. J. and P. K. Kitadinis, Comparison of gausian conditional mean and kriging in the geostatistical solution of the inverse problem, *Water Resources Research,* **21**(6), 825-836, 1985b.

Hsu, N. S. and W. Yeh, Optimum experimental design for parameter identification in groundwater hydrology, *Water Resources Research ,* **25**(5), 1025-1040, 1989.

Husain, T., Hydrologic uncertainty measure and network design, *Water Resources Bulletin,*

25(3), 527-534, 1989.

IAHS, *Monitoring to Detect Changes in Water Quality, Proceedings of the Budapest Symposium, July, 1986*, Publication No. 157, IAHS Press, 1986.

Ingber, L., Very fast simulated re-annealing, *Mathematical and Computer Modelling,* **12**(8), 967-973, 1989.

Ingber, L., Simulated annealing: practice versus theory, *Mathematical and Computer Modelling,* **18**(11), 29-57, 1993 .

Ingber, L. and J. K. Wilson, Statistical mechanics of financial markets: Exponential modifications to Black-Scholes, *Mathematical and Computer Modelling,* **31**(8-9), 167-192 , 2000.

James, B. R. and S. M. Gorelick, When enough is enough: the worth of monitoring data in aquifer remediation design, *Water Resources Research,* **30**(12), 3499-3513, 1994.

Johnson, D. S., C. R. Aragon, L. A. McGeoch and C. Schevon, Optimization by simulated annealing: an experimental evaluation, Part I, Graph partitioning, *Operations Research,* **37**(6), 865-892 , 1991a.

Johnson, D. S., C. R. Aragon, L. A. McGeoch and C. Schevon, Optimization by simulated annealing: an experimental evaluation, Part II, Graph coloring and number partitioning, *Operations Research,* **39**(3), 378-406, 1991b.

Johnson, V. M. and L. L. Rogers, Accuracy of neural network approximators in simulation-

optimization, *Journal of Water Resources Planning and Management-ASCE ,* **126**(2), 48-56, 2000.

Johnson, V. M. and L. L. Rogers, Applying soft computing methods to improve the computational tractability of a subsurface simulation-optimization problem, *Journal of Petroleum Science and Engineering ,* **29**(3-4), 153-175, 2001.

Journel, A.*, Geostatistics for the Environmental Sciences, An Introduction*, Applied Earth Sciences Department, Stanford University, Project No. CR 811893, California, U.S.A., 1987.

Journel, A., *Fundamentals of Geoestatistics in Five Lessons, Volume 8, Short Course in Geology,* American Geophysical Union, Washington, DC, U.S.A., 1989

Journel, A. and Ch. Huijbregts, *Mining Geostatistics*, Academic Press, New York, U.S.A., 1978.

Journel, A. G., Non-parametric estimation of spatial distribution, *Mathematical Geology,* **15**(3), 445-465, 1983.

Kirkpatrick, S., Jr. C. D. Gellat and M. P. Vecchi, Optimization by simulated annealing, *Science,* **220**(4598), 671-680, 1983.

Knopman, D. S. and C. I. Voss, Multiobjective sampling design for parameter estimation and model discrimination in groundwater solute transport, *Water Resources Research,* **25**(10), 2245-2258, 1989.

Kuo, S. F., C. W. Liu and S. K. Chen, Comparative study of optimization techniques for irrigation project planning, *Journal of the American Water Resources Association,* **31**(9), 59-73,

2003 .

Kuo, S. F., C. W. Liu and G. P. Merkley, Application of the simulated annealing method to agricultural water resource management, *Journal of Agricultural Engineering Research ,* **80**(1), 109-124 , 2001.

Laarhoven, P. J. M. v. and E. Aarts, *Simulated annealing: Theory and applications*, Reidel, Dordrecht, 1987.

Lack, T. J.*, Draft annual summary report 1996,* edited by ETC/IW - European Topic Centre on Inland Waters, WRc Ref. P017/97/2, 1997.

Lebel, T., G. Bastin, C. Obled and J. D. Creutin, On the accuracy of areal rainfall estimation: a case study, *Water Resources Research,* **23**(11), 2123-2134, 1987.

Lee, S.-I. and P. K. Kitadinis, Optimal estimation and scheduling in aquifer remediation with incomplete information, *Water Resources Research,* **27**(9), 2203-2217, 1991.

Lee, Y.-M. and J. H. Ellis, Comparison of algorithms for nonlinear integer optimization: application to monitoring network design, *Journal of Environmental Engineering,* **122**(6), 524-531, 1996.

Lenton, R. L. & Rodríguez-Iturbe, I.*, On the collection, the analysis, and the synthesis of spatial rainfall data,* edited by Lenton R L and Rodríguez-Iturbe I, Ralph M. Parsons Lab. for Water Resources and Hydrdynamics, Dep. of Civil Eng., Mass. Inst. of Technology, Tech. Rep. 194, Cambridge, U.S.A., 1974.

Lettenmaier, D. P. and S. J. Burges, Design of trend monitoring networks, *Journal of Environmental Engineering- ASCE,* **103**(5), 785-802, 1977.

Loaiciga, H. A., An optimization approach for groundwater quality monitoring network design, *Water Resources Research,* **25**(8), 1771-1782, 1989.

Loaiciga, H. A., R. J. Charbeneau, L. G. Everett, G. E. Fogg, B. F. Hobbs and S. Rouhani, Review of ground-water quality monitoring network design, *Journal of Hydraulic Engineering,* **118**(1), 11-37, 1992.

Lundy, M. and A. Mees, Convergence of an annealing algorithm, *Mathematical Programming,* **34**(1), 111-124, 1986.

Markus, A., W. J. Welch and J. Sacks, Design and analysis for modeling and predicting spatial contamination, *Mathematical Geology,* **31**(1), 1-22, 1999.

Marryott, R. A., D. E. Dougherty and R. L. Stollar, Optimal groundwater management, 2. Application of simulated annealing to a field-scale contamination site, *Water Resources Research,* **29**(4), 847-860, 1993.

Massmann, J. and R. A. Freeze, Groundwater contamination from waste management sites: the interaction between risk-based engineering design and regulatory policy, 1. Methodology, *Water Resources Research,* **23**(2), 351-367, 1987a.

Massmann, J. and R. A. Freeze, Groundwater contamination from waste management sites: the interaction between risk-based engineering design and regulatory policy, 2. Results, *Water Resources Research,* **23**(2), 368-380, 1987b.

Massmann, J., R. A. Freeze, L. Smith, T. Sperling and B. James, Hydrologeological decision analysis, 2: results, *Ground Water,* **29**(4), 536-548, 1990.

Matalas N. C.andM. B. Fiering , Water resource system planning, in *Climate, Climatic Change and Water Supply,* edited by National Research Council, pp. 64-72, National Academy of Sciences, Washington, D.C., 1977.

Matheron, G., Principles of geostatistics, *Economical Geolpgy,* **58**, 1246-1266, 1963.

Matheron, G., *Les variables regionalisées et leur estimation*, Masson et Cie, Paris, France, 1965.

Matheron, G., La théorie des variables régionalizées, et ses applications, *Les Cahiers du Centre de Morphologie Mathématique de Fontainbleau,* **Fascicule 5***,* 1970.

McBratney, R. W., R. Webster and T. M. Burgess, The design of optimal sampling schemes for local estimation and mapping of regionalized variables - I, *Computers and Geosciences,* **7**(4), 331-334, 1981a.

McBratney, R. W., R. Webster and T. M. Burgess, The design of optimal sampling schemes for local estimation and mapping of regionalized variables - II, *Computers and Geosciences,* **7**(4), 335-365, 1981b.

McKinney D. C.andD. P. Loucks, Uncertainty analysis methods in groundwater modeling, in *Computational Methods in Surface Hydrology,* edited by G. Gambolati, A. Rinaldo, C. Brebbia, W. Gray and G. Pinder, pp. 479-485, Springer-Verlag, Berlin, 1990.

McKinney, D. C. and D. P. Loucks, Network design for predicting groundwater contamination, *Water Resources Research,* **28**(1), 133-147, 1992.

McLaughlin D. B.andW. D. Graham, Design of cost-effective programs for monitoring groundwater contamination, in *Integrated Design of Hydrologic Networks, No. 158,* edited by IAHS, pp. 31-245, IAHS Publ., 1986.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, Equation of state calculations by fast computing machines, *The Journal of Chemical Physics,* **21**(6), 1087-1092, 1953.

Meyer, P. D. and B. Downey, A method for locating wells in a groundwater monitoring network under conditions of uncertainty, *Water Resources Research,* **24**(8), 1277-1282, 1988.

Meyer, P. D., A. J. Valocchi and J. W. Eheart, Monitoring network design to provide initial detection of groundwater contamination, *Water Resources Research,* **30**(9), 2647-2659, 1994.

Moss, M., *Integrated Design of Hydrological Networks,* IAHS Publication No. 158, IAHS Press, Institute of Hydrology, Wallingford, Oxfordshire, 1986.

Nixon, S. C*., European freshwater monitoring network design. Topic report No 10/1996*, edited by Nixon S C, European Environment Agency, Copenhagen, 1996.

Nixon, S. C., Rees, Y. J., Gendebien, A. & Ashley, S. J*., Requirements for water monitoring, Topic report No 1/1996,* edited by Nixon S C, Rees Y J, Gendebien A and Ashley S J, European Environment Agency, Copenhagen, 1996.

Oliver, M. A. and R. Webster, A geostatistical basis for spatial weighting in multivariate classification, *Mathematical Geology,* **21**(1), 15-35, 1989.

Onbasoglu, E. and L. Ozdamar, Parallel simulated annealing algorithms in global optimization, *Journal of Global Optimization,* **19**(1), 27-50, 2001.

Otten, R. H. and L. P. van Ginneken, *The annealing algorithm*, Kluwer Academic Publishers, U.S.A., 1989.

Palmer, R. N. and M. C. McKenzie, Optimization of water quality monitoring networks, *Journal of Water Resources Planning and Management,* **111**(4), 478-493, 1985.

Pardo-Igúzquiza, E., Optimal selection of number and location of rainfall gauges for areal rainfall estimation using geostatistics and simulated annealing, *Journal of Hydrology,* **210**(1-4), 206-220, 1998.

Patel, N. R., R. L. Smith and Z. B. Zabinsky, Pure adaptive search in Monte Carlo optimization, *Mathematical Programming,* **43**(3), 317-328, 1988.

Paralta, E., *Hidrogeologia e Modelação Estocástica da Contaminação por Nitratos do Aquífero Gabro-diorítico da Região de Beja*, MSc Thesis, Technical University of Lisbon, IST, Lisbon, 2001.

Popov, M. and S. He, Identification of a transient electric dipole over a conducting half space using a simulated annealing algorithm, *Journal of Geophysical Research-Atmospheres,* **105**(D16), 20821-20831, 2000.

Reed, P., B. Minsker and A. J. Valocchi, Cost-effective long-term groundwater monitoring design using a genetic algorithm and global mass interpolation, *Water Resources Research,* **36**(12), 3731-3741, 2000.

Rinnooy Kan, A. H. G. and G. T. Timmer, Stochastic global optimization methods; Part I: clustering methods, *Mathematical Programming,* **39**(1), 27-56 , 1987a.

Rinnooy Kan, A. H. G. and G. T. Timmer, Stochastic global optimization methods; Part II: multi level methods, *Mathematical Programming,* **39**(1), 57-78, 1987b.

Rinnooy Kan A. H. G.and G. T. Timmer , Global optimization, in *Handbooks in Operations Research and Management Science, Volume 1: Optimization,* edited by G. L. Nemhauser, A. H. G. Rinnooy Kan and M. J. Todd, pp. 631-662, North-Holland, Amsterdam, The Netherlands, 1989.

Rodríguez-Iturbe, I. and J. M. Mejía, The design of rainfall networks in time and space, *Water Resources Research,* **10**(4), 713-728, 1974a.

Rodríguez-Iturbe, I. and J. M. Mejía, On the transformation of point rainfall to areal rainfall, *Water Resources Research,* **10**(4), 729-736, 1974b.

Romeijn, H. E. & Smith, R. L.*, Simulated annealing and adaptive search in global optimization*, Management Report 151, edited by Romeijn H E and Smith R L, Rotherdam School of Management Rotherdam, The Netherlands, 1993.

Romeijn, H. E. and R. L. Smith, Simulated annealing for constrained global optimization, *Journal of Global Optimization,* **5**(2), 101-126, 1994.

Rouhani, S., Variance reduction analysis, *Water Resources Research,* **21**(6), 837-846, 1985.

Rouhani, S. and M. B. Fiering, Resilience of a statistical sampling scheme, *Journal of Hydrology,* **89**(1-2), 1-11, 1986.

Rouhani, S. and T. Hall, Geostatistical schemes for groundwater sampling, *Journal of Hydrology,* **103**(1-2), 85-102, 1988.

Rubinstein, R. Y., *Simulation and the Monte Carlo method*, Wiley, New York, 1981.

Sacks J.and S. Schiller, Spatial designs, in *Statistical Decision Theory and Related Topics IV,* vol 2, edited by S. S. Gupta and J. O. Berger, pp. 385-399, Springer-Verlag, New York, 1988.

Sanders, T. G., R. C. Ward, J. C. Loftis, T. D. Steel, D. D. Adrian and V. Yevjevich, *Design of networks for monitoring water quality*, Water Resources Publications, Littleton, Colorado, 1983.

Scheibe, T. D. and Y. J. Chien, An evaluation of conditioning data for solute transport prediction, *Ground Water,* **41**(2), 128-141, 2003.

Scheidleder, A., Grath, J., Götz, B., Kralik, M. & Weber, K.*, Pilot implementation EUROWATERNET - groundwater - State of groundwater in selected groundwater bodies with reference to nitrogen,* Technical report No 39*,* edited by Scheidleder A, Grath J, Götz B, Kralik M and Weber K, European Environment Agency, Copenhagen, 2000.

Scheidleder, A., Grath, J., Winkler, G., Stark, U., Koreimann, C., Gmeiner, C., Nixon, S., Casillas, J., Gravense, P., Leonard, J. & Elvira, M.*, Groundwater quality and quantity in*

Europe, edited by European Environment Agency, European Environment Agency Copenhagen, 1999.

Shafike, N. G., L. Duckstein and T. Maddock, Multicriterion analysis of groundwater contamination management, *Water Resources Bulletin,* **28**(1), 33-43, 1992.

Shannon, C. E., A mathematical theory of communication, *Bell System Technical Journal,* **27**, 379-423, 1948.

Shannon, C. E. and W. Weaver, *The Mathematical theory of communication*, The University of Illinois Press, Urbana, Illinois, U.S.A., 1949.

Sharp, W. E., A topographically optimum water sampling plan for rivers and streams, *Water Resources Research,* **7**(6), 1641-1646, 1971.

Shreve, R. L., Infinite topologically random channel networks, *Journal of Geology,* **75**(2)**,** 178-186, 1967.

Siarry, P., Enhanced simulated annealing for globally minimizing functions of many-continuous variables, *ACM Transactions on Mathematical Software,* **23**(2), 209-228, 1997.

Skaggs, R. L., L. W. Mays and L. W. Vail, Application of enhanced annealing to ground water remediation design, *Journal of the American Water Resources Association,* **37**(4), 867-875, 2001a.

Skaggs, R. L., L. W. Mays and L. W. Vail, Simulated annealing with memory and directional search for ground water remediation design, *Journal of the American Water Resources*

Association, **37**(4), 853-866, 2001b.

Skalski, J. R. and D. H. McKenzie, A design for aquatic monitoring programs, *Journal of Environmental Management,* **14**(3), 237-251, 1982.

Smith, R. L., Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions, *Operations Research,* **32**(6), 1296-1308, 1984.

Solis, F. J. and R. J. B. Wets, Minimization by random search techniques, *Mathematics of Operations Research,* **6**(1), 19-30, 1981.

Storck, P., J. W. Eheart and A. J. Valocchi, A method for the optimal location of monitoring wells for detection of groundwater contamination in three-dimensional heterogenous aquifers, *Water Resources Research ,* **33**(9), 2081-2088, 1997.

Sullivan, J. A., *Non-parametric estimation of spatial distribution*, PhD Thesis, Stanford University, Stanford, 1984.

Thompson, S. K., *Sampling*, John Wiley & Sons, Inc., New York, 1992.

Tiedman, C. and S. Gorelick, Analysis of uncertainty in optimal groundwater contaminant capture design, *Water Resources Research,* **29**(7), 2139-2153, 1993.

Todd, D. K., Timlin, R. M., Schmidt, K. D. & Everett, L. G., *Monitoring ground water quality: Monitoring methodology*, United States Environmental Agency, Las Vegas, USA, 1976.

Törn, A. A., Cluster analysis using seed points and density determined hyperspheres with an

application to global optimization, paper presented at *Proceedings of the International Conference on Pattern Recognition,* Coronado, California, 1976.

Törn A. A., A search clustering approach to global optimization, in *Towards Global Optimization 2,* edited by L. C. W. Dixon and G. P. Szegö, pp. 49-62, North-Holland, Amsterdam, The Netherlands, 1978.

Wagner, B. J. and S. M. Gorelick, Optimal groundwater quality management under parameter uncertainty, *Water Resources Research,* **23**(7), 1162-1174, 1987.

Wagner, B. J. and S. M. Gorelick, Reliable aquifer remediation in the presence of spatially variable hydraulic conductivity: from data to design, *Water Resources Research,* **25**(10), 2211-2225, 1989.

Wang, M. and C. Zheng, Ground water management optimization using genetic algorithms and simulated annealing: Formulation and comparison, *Journal of the American Water Resources Association ,* **34**(3), 519-530, 1998.

Ward, R. C. and J. C. Loftis, Establishing statistical design criteria for water quality monitoring systems: review and synthesis, *Water Resources Bulletin,* **22**(5), 759-767, 1986.

Whitfield, P. H., Goals and data collection designs for water quality monitoring, *Water Resources Bulletin,* **24**(4), 775-780, 1988.

WMO*, Advances in water quality monitoring - Report of a WMO regional workshop (Vienna, 7-11 March 1994)*, edited by World Meteorological Organization, Geneva, Switzerland, 1994.

Woldt, W. and I. Bogardi, Ground water monitoring network design using multiple criteria decision making and geostatistics, *Water Resources Bulletin,* **28**(1), 45-62, 1992.

Wood, E. F. and D. B. McLaughlin, Groundwater monitoring well redesign, paper presented at *Conference on Petroleum Hydrocarbons and Organic Chemicals in Groundwater: Prevention, Detection, and Restoration,* 1984.

Yfantis, E. A., G. T. Flatman and J. V. Behar, Efficiency of kriging estimation for square, triangular, and hexagonal grids, *Mathematical Geology,* **19**(3), 183-205, 1987.

Zabinsky, Z. B. and R. L. Smith, Pure adaptive search in global optimization, *Mathematical Programming,* **53**(3), 323-338, 1992.

APPENDIX A: FORTRAN 90 OPTIMIZATION CODES

CODE FOR MONITORING NETWORK OPTIMIZATION

```
         Program crossvar

! Programa para optimiza‡Æo de redes de amostragem por simulated annealing

! O programa l^ um conjunto de localiza‡äes da rede disponivel, determina
! a fun‡Æo objectivo(dO) e a varia‡Æo de temperatura(dt); para cada passo
! de temperatura sÆo executados nover ciclos, ou nlimit passos aceites (dimi-
! nui‡Æo da fun‡Æo objectivo.)
!
!  npontos: n° de pontos a experimentar
!  ni: n° de dados
!  nover: n° de ciclos de arrefecimento
!  nlimit: n° de passos aceites
!  t: temperatura inicial
!  tmin temperatura mínima
!  rep: repetições até aceitação
!  n: n° de coordenadas finais (ou n° de pontos novos a incluir)
!
!  filename: ficheiro de parâmetros
!  fout: as coordenadas dos pontos optimos e os resultados são escritos neste ficheiro
!  fokin: ficheiro com os dados de entrada para o programa de krigagem (GSLIB)
!  fokout ficheiro com os resultados da krigagem (valor estimado e variancia)
!
!
!  PARA MINIMIZAR OU MAXIMIZAR A REDUNDANCIA DEVE ALTERAR-SE A ROTINA COST
!
        Use DFLIB
              USE DFPORT

        Implicit none


!
*******************************************************************************
!
!               DEFINICOES
!
```

```
!  ni                                  ! No. de pontos da rede ja existente
!  npontos                             ! No. total de novos pontos disponiveis
!  n                                   ! No. de pontos a incluir
!  nover                               ! No. de ciclos de arrefecimento
!  nlimit                              ! No. de ciclos a temp. const.
!  t                                   ! Temp. inicial
!  tmin                             ! Temp. minima
!  rep                                 ! No. de repeticoes de delta C const. necessarios para
baixar a temp.
!  tfactr                              ! Factor de arrefecimento
!  stopsum                             ! No. de repetições com F.O. constante a difs temps
!  fout (4)                    ! Fich com os resultados da optimizacao
!  fokin (5)                   ! Fich com coordenadas para a krigagem
!  fokout (6)                  ! Fich com os resultados da krigagem
!  cross (18)                  ! Fich com os resultados da cross-validation
!  monitor1                    ! Fich para analise de sensibilidade - a partir de temp5.tmp
!  monitor2                            ! Fich para analise de sensibilidade - a partir de
temp6.tmp
!  variancia_inicial   ! Delta C inicial
!  i,j,k,a                             ! contadores de ciclos locais
!  count                               ! contador de vectores iguais - subroutine vectores
!  count_nlimit         ! contador de ciclos a temperatura constante - compara com nlimit
!  count_limit_internal ! constante igual a nover*nlimit
!  limit_counter        ! contador de ciclos totais - compara com count_limit_internal
!  count_sum            ! contadore de repetiçoes de medias em corridas consecutivas -
!                                                          compara com stopsum
!  count_rep                   ! contador de repetições da F.O. - compara com rep
!  hour, minute, second, p100second  ! variáveis de tempo - função gettim()       - usadas
na
!                                                    construção das variaveis aleatorias ra,
dummy1 e dummy2
!  x,y                                 ! matrizes de coordenadas dos novos pontos a analisar

!  xx,yy                               ! matrizes de coordenadas com todos os pontos
!  xn,yn                               ! matrizes de coordenadas com os novos pontos a
analisar mais os
!                                           !    pontos já existentes
!  coordenadas_iniciais ! vector usado para filtrar xx e yy - obter apenas os pontos
iniciais
!  coordenadas_totais   ! vector usado para filtrar xx e yy - obter todos os pontos
!  coordenadas_a_testar ! vector usado para filtrar xx e yy - obter todos os pontos a
testar
!  timesincejanfirst1970  ! tempo decorrido em segundos desde 1/1/1970 - funcao Time()
!  sumdif                              ! valor do somatorio das =/  series temporais -
subroutine minsumdif
!  variance                            ! igual a den, valor da funcao objectivo no passo
actual
!  Oldvariance                 ! igual a oldden, valor da funcao objectivo no passo anterior
!  Objective                   ! Diferença entre den e oldden (delta C)
!  summ                                ! somatorio de variance (C) para um ciclo de
temperatura - para calculo
!                                                    da media e variancia durante o ciclo
!  summ2                               ! somatorio do quadrado de summ - para calculo da
variancia durante
!                                                    um ciclo de temperatura
!  Oldsum                              ! valor de summ guardado entre corridas no ficheiro
temp100.tmp
!  Oldsum2                             ! valor de summ2 guardado entre corridas no ficheiro
temp100.tmp
!  maxden                              ! valor máximo da funcao objectivo - usado para
normalizar o valor de C
!  maxsumdif                   ! valor maximo da diferenca das series (sumdif) - usado para
normalizar
!                                                           o valor de C
!  krigvarm                            ! valor da variancia de krigagem - resultado de kt3d
!  oldkrigvar                  ! valor da sk2 correspondente a oldden
!  ans                                 ! variavel logica com valor verdadeiro quando
objective <= 0
!  signall                             ! variavel logica com valor verdadeiro quando
count_rep = rep
!  filename    (3)                     ! nome do ficheiro de parametros deste programa
```

```
!  n_iteracoes                ! numero de iteracoes consecutivas deste programa
!  iter                              ! contador do numero de iteracoes consecutivas deste
programa
!  oldtime                           ! marcador do tempo na primeira corrida de cada
iteracao
!  olybettercounter           ! contador semelhante ao count_rep, mas em são contabilizadas
apenas
!                                          !    as alterações que produzam resultados
melhores (e não <=)
!  maxmin                            ! interruptor: 0: min sk2 e max redundancia; 1: min
sk2 e min redund.
!
! As restantes variáveis são variáveis locais ou dummy.
!
! FICHEIROS TEMPORARIOS
!
! temp1.tmp           ! (unit=7) t, count_nlimit, limit_counter, count_rep
! temp2.tmp           ! (8) oldden, summ, summ2, maxkrigvar, maxsumdif,
oldkrigvar,oldsumdif
! temp3.tmp           ! (9) vector(i)  - da última corrida
! temp4.tmp           ! (10) vector(i) - melhor solucao ate ao momento
! temp5.tmp           ! (20) limit_counter, oldvariance, krigvarm, sumdif, &
!                       timesincejanfirst1970-oldtime
! temp6.tmp           ! (21) limit_counter, (summ/count_rep), summ2/count_rep-
(summ/count_rep)**2, &
!                       timesincejanfirst1970-oldtime
! iter.tmp            ! (87) iter, oldtime
! temp100.tmp  ! (100) Oldsum, Oldsum2, count_sum
! stop.stp            ! (38) - vazio - serve apenas para parar ao fim das n_iteracoes
! limit00.lmt  ! (36) - vazio - serve apenas para exportar resultados ao fim do ciclo de t
! fim.fim             ! (34) - vazio - serve para indicar fim de simulação (iteracao)
!
!
! O resultado das corridas e gravado nos ficheiros monitor.out e monitor2.out
!
!
*****************************************************************************************


                Integer::nover, nlimit,  openstatus, rep, count_limit_internal,&
                           limit_counter, count_sum, stopsum, count_rep,&
                         npontos, n, i,j,k, a, ni,count_nlimit, count, nserie,
ndiferencial,&
                                  n_iteracoes, iter, onlybettercounter


                Integer(4):: timesincejanfirst1970, oldtime

                Integer(2):: hour, minute, second, p100second, maxmin


          Real(4), Allocatable, Dimension(:):: x,y, coordenadas_iniciais, &
                         coordenadas_a_testar, coordenadas_totais, xx, yy, quociente,&
                                xsolucao, ysolucao

!               Real(4), Allocatable, Dimension(:)::yn,xn



                Integer, allocatable, Dimension(:):: xk1,k1, solucao, xk1solucao

!               xk2, xk3, xk4, k2, k3, k4


!               Integer, allocatable, Dimension(:)::  k1n, k2n, k3n, k4n



          Real(4)::  tfactr, tmin,ra, variancia_inicial, sumdif,&
                         variance, Oldvariance, objective,t,&
```

```
                            hx01, hx02, hx03, hx04,  hx06,maxden, krigvarm,&
                               oldkrigvar, summ,oldsum, maxkrigvar, sk2_coef


!               Real(8):: summ,oldsum
                Logical:: ans,signall

          Character(12):: filename, fout, fokin, fokout, cross, monitor, monitor2



                 filename='optivar.par'

          Open(unit=3,file=filename, iostat=openstatus)
          Open(unit=8, file='temp2.tmp', iostat=openstatus)
!          Open(unit=33, file='aaa.aaa', iostat=openstatus)
                Open(unit=87,file='iter.tmp', iostat=openstatus)
          If (openstatus>0) stop '****** Ficheiro invalido ******'


          Read (3,*)


           read (3,*) ni                                        ! No. de pontos da rede ja
existente
                  read(3,*) npontos                             ! No. total de novos pontos
disponiveis
                  read(3,*) n                                   ! No. de pontos a incluir
                  read (3,*) nover                              ! No. de ciclos de
arrefecimento
                  read (3,*) nlimit                             ! No. de ciclos a temp. const.
                  read (3,*) t                                  ! Temp. inicial
                  read (3,*) tmin                               ! Temp. minima
                  read(3,*) rep                                        ! No. de repeticoes de
delta C const. necessarios para baixar a temp.
                  read(3,*) tfactr                              ! Factor de arrefecimento
                  read(3,*) stopsum                             ! No. de repetições com F.O.
constante a difs temps
                  read (3,*) fout                               ! Fich com os resultados da
optimizacao
                  read (3,*) fokin                              ! Fich com coordenadas para a
krigagem
                  read (3,*) cross                              ! Fich com as series temporais
                  read(3,*) variancia_inicial ! Delta C inicial
                  read(3,*) n_iteracoes                         ! Numero de iteracoes
consecutivas deste programa
                  read(3,*) maxmin                              ! 1: maxmin; 0; minmin
                  read(3,*) sk2_coef                 ! coeficientes da funcao custo




! Se o numero de iteracoes deste programa iguala o numero pre-definido, entao o programa e
! interrompido e e criado um ficheiro (stop.stp) que da sinal a a.bat para terminar o
processo



                  Allocate (coordenadas_iniciais(ni))
                  Allocate (coordenadas_totais(ni+npontos))
                  Allocate (coordenadas_a_testar(npontos))



          Open(unit=4,file=fout, iostat=openstatus)
                Open(unit=7, file='temp1.tmp', status='OLD', iostat=openstatus)
                Open(unit=18, file='cross.dat', iostat=openstatus)
                Open(unit=10, file='temp4.tmp',status='OLD', iostat=openstatus)
                Open(unit=20, file='temp5.tmp', iostat=openstatus)
```

```
                Open(unit=21, file='temp6.tmp', iostat=openstatus)
                Open(unit=100, file='temp100.tmp', iostat=openstatus)
                If (openstatus>0) stop '****** Ficheiro invalido ******'
                open (unit=5, file=fokin, status='OLD', iostat=openstatus)
                If (openstatus>0) stop '****** Ficheiro invalido ******'
          open(unit=9, file='temp3.tmp', iostat=openstatus)
                If (openstatus>0) stop '****** Ficheiro invalido ******'

          Allocate (x(npontos))
          Allocate (y(npontos))

                Allocate (k1(npontos))
!          Allocate (k2(npontos))
!               Allocate (k3(npontos))
!         Allocate (k4(npontos))

!          Allocate (xn(n))
!          Allocate (yn(n))
!               Allocate (k1n(n))
!          Allocate (k2n(n))
!               Allocate (k3n(n))
!          Allocate (k4n(n))


                Allocate (xx(npontos+ni))
                Allocate (yy(npontos+ni))

                Allocate (xk1(npontos+ni))
!               Allocate (xk2(npontos+ni))
!               Allocate (xk3(npontos+ni))
!               Allocate (xk4(npontos+ni))

                Allocate(solucao(n+ni))
                Allocate(xsolucao(n+ni))
                Allocate(ysolucao(n+ni))
                Allocate(xk1solucao(n+ni))


!Lê o ficheiro com as coordenadas de todos os pontos disponíveis

                s799: do i=1,npontos+ni


             Read (3,*) xx(i), yy(i), xk1(i)

!                       , xk2(i), xk3(i), xk4(i)



           end do s799

                 read(87,*)
                 read (87,*) iter, oldtime



!                      write(*,*) 'iter: ', iter


                 if(iter.gt.n_iteracoes) then

                      open(unit=88,file='solution.tmp')


                      do i=1,ni

                       solucao(i)=i

                      end do
```

```
                        do i=ni+1,n
                                read(10,*) solucao(i)
                                solucao(i)=solucao(i)+ni
                        end do

                        xsolucao = xx(solucao)
                        ysolucao = yy(solucao)
                        xk1solucao = xk1(solucao)

                        do i=1,n+ni
                                write(88,*) xsolucao(i), ysolucao(i), xk1solucao(i)
                        end do
                      open(unit=38,file='stop.stp')
!                       goto 3001
                        goto 7
                    end if


!Coordenadas totais
           do i=1, ni+npontos
                    coordenadas_totais(i)=i
                  end do

! Coordenadas iniciais
            do i=1, ni
                     coordenadas_iniciais(i)=i
                    end do
! Coordenadas a testar
                  do i=1,npontos
                    coordenadas_a_testar(i)=i+ni  !As primeiras ni linhas são os pontos fixos


                  end do


! Caso seja a primeira corrida
                  if (iter.eq.0) then
                        ifif02: do i=1,5
                                read(5,*)
                        end do ifif02
```

```
                        ifif01: do i=1, n+ni

                               write(5,*) xx(i), yy(i), xk1(i)

                        end do ifif01

                        do i=1,n

                               write(9,*) i
                               write(10,*) i

                        end do

                        iter=1

                        rewind(87)

                        write(87,*)
                        write (87,*) iter, oldtime

                        goto 9999

                end if


              x=xx(coordenadas_a_testar)
                        y=yy(coordenadas_a_testar)
                        k1=xk1(coordenadas_a_testar)
!                       k2=xk2(coordenadas_a_testar)
!                       k3=xk3(coordenadas_a_testar)
!                       k4=xk4(coordenadas_a_testar)


! teste do vector a eliminar quando correcto

!              do i=1, ni+npontos

!                         write(33,*) 'xx(i), yy(i), '
!                         write(33, *) xx(i),  yy(i)

!              end do

!           do i=1, npontos

!                         write(33,*) 'x(i), y(i), '
!                         write(33, *) x(i),  y(i)

!              end do


                  rewind(7)
                               read(7,*)
                               read(7,*) t, count_nlimit, limit_counter, count_rep,
onlybettercounter


!                              write(*,*) 't, count_nlimit, limit_counter, count_rep,
onlybettercounter'
!                              write(*,*) t, count_nlimit, limit_counter, count_rep,
onlybettercounter


                               timesincejanfirst1970 = Time()


                   if(limit_counter.lt.1) then

!                               write(*,*) 'ln 301'

                                  oldtime=Time()
```

```
                                    rewind(20)

                                    write(20,*) 'Corridas, F.O., krig var,  tempo'

                                    rewind(21)

                                    write(21,*) 'Corridas, media de F.O., var F.O.,
tempo'

                                                                          rewind(87)
                                    write(87,*) 'iter, oldtime'
                                    write(87,*) iter, oldtime

!                                  s21: do a=1,n
!                       xn(a)=a
!                       yn(a)=a
!                                    k1n(a)=a
!                                    k2n(a)=a
!                                    k3n(a)=a
!                                    k4n(a)=a
!                       enddo s21

                                 else

                                    goto 100

                                   end if



100                                count_limit_internal=nover*nlimit  !n° de ciclos
totais

                                 limit_counter=limit_counter+1      !contador de ciclos
totais

                     count_nlimit=count_nlimit+1          !contador de ciclos a t const.



!                                  write(*,*) 'ln 224'

                                 call cost(n, objective,variance, Oldvariance, ni, fokout)


                   call metrop(objective,variance,oldvariance,t, ans, signall)


                   call vectores(n, ni, npontos, x, y, k1)

!                                , k2, k3, k4)


                                    rewind(100)
                                   read(100,*)
                                   read(100,*) Oldsum,  count_sum


!                                         write(*,*) 'Oldsum,  count_sum'

!                                         write(*,*) Oldsum,  count_sum


!                                         write(*,*) 'timesincejanfirst1970-oldtime'
!                                         write(*,*) timesincejanfirst1970-oldtime
```

```
! Enquanto nao se atingir o limite de iteracoes, faz

                  if (t>tmin.and.limit_counter.lt.&
                                          count_limit_internal.and.count_sum.lt.stopsum)
then


! Se o numero de repeticoes da F.O. a t const. ou o numero maximo de iteracoes a t const
! são atingidos, ou stopsum não é tingido, então: actualiza a temperatura



                                       if(count_nlimit.eq.nlimit.or.signall) then


!                                      if(summ/count_rep.lt.oldsum+(summ/count_rep-
oldsum)+oldsum/10000.&
!
and.summ/count_rep.gt.oldsum+(summ/count_rep-oldsum)&
!                                                        -oldsum/10000) then

                                                 if(summ/count_rep.lt.oldsum+oldsum/1e6.&
                                                     and.summ/count_rep.gt.oldsum-oldsum/1e6)
then


                                                          count_sum=count_sum+1

                                                   rewind(100)
                                                   read(100,*)
                                                   write(100,*) oldsum,  count_sum

!                                                  write(*,*) 'count_sum ', count_sum


                                               else

                                                    count_sum = 0

                                                    if(count_rep.eq.0) then

                                                      count_rep=1

                                                    end if

                                                    rewind(100)
                                                    read(100,*)
                                                    write(100,*) (summ/count_rep), count_sum

!                                                   write(*,*) 'count_sum ', count_sum

!                                                   write(33,*) 'se condicao F'
!                                                   write(33,*) ' < ',oldsum+oldsum/1e6
!                                                write(33,*) summ/count_rep
!                                                write(33,*) ' > ', oldsum-oldsum/1e6

                                               end if



! Escreve para um ficheiro de resultados os seguintes dados:
! No. da iteração
! Temperatura
! Dimensao da cadeia de Markov
! delta C actual
! Média de delta C
! Variância de delta C

                                       Write(20,*) limit_counter, oldvariance, &
```

```
                                    oldkrigvar,  &
                                    timesincejanfirst1970-oldtime


                                    write(21,*)
limit_counter,count_rep,onlybettercounter,&
                                                    (summ/count_rep), t



                                open(unit=36,file='limit00.lmt')

!                                    summ=0.0


                                    t=t*tfactr

                                    count_nlimit=0
                                    count_rep=0
                                    onlybettercounter=0



                                rewind(8)
                                read(8,*)
                                read(8,*) hx01, hx02, hx04,  hx06

                                hx02=0.0


                                rewind(8)
                                read(8,*)

                                write(8,*) hx01, hx02,  hx04,  hx06

!                               write(33,*) hx01, summ,  hx04,  hx06


                                      goto 7


                                    else

                                      goto 7

                                          end if


                                    else

                                     goto 6


                  end if


!                               if (t<tmin.or.limit_counter.ge.&
!                                        count_limit_internal) then
!
!                                  goto 6

!                                  else

!                                      goto 7

!                               end if
```

```
6                write (*,*)

                              if(limit_counter.eq.count_limit_internal.and.t.gt.tmin) then

                                    write(*,*) ' Temp. minima não atingida'
                                       write(4,*) 'Temp. minima não atingida: t>tmin'

                end if


                                     write(4,*) 'T= ',t


!                s505: do a=1,n
!                  write(4,*) 'x= ',xn(a),'    y= ', yn(a)
!                enddo s505


                              open(34,file='fim.fim')


! Actualiza o contador de iteracoes

                             iter = iter + 1


7                            rewind(87)
                             write(87,*) 'iter, oldtime'
                             write(87,*) iter, oldtime

                              rewind(unit=7)
                              write(7,*) 't //', 'rep. t const. //',' repet.totais //','
count_rep'
                              write(7,*)  t, count_nlimit, limit_counter, count_rep,
onlybettercounter


            close(7)
                       close(4)
            close(8)
                       close(18)
                       close(10)
                       close(3)
            close(33)
                       close(34)
                       close(87)
                       close(38)
                       close(100)


         Contains

! Subrotina para determina‡Æo da fun‡Æo objectivo. DÆo entrada os vectores de
! localiza‡Æo xnn e ynn, com dimensÆo n (n£mero definido … priori de esta‡äes
! a serem utilizadas. Sai o valor estimado para a itera‡Æo, den
! No caso de se utilizar a vari ncia de krigagem como indicadora do custo -
! ver Pardo-Ig£zquiza (1998)


       Subroutine cost(nn, object, den,oldden, nni, fokoutn)

           Implicit none

           Integer, intent(in):: nn, nni

           Real(4), intent(out):: object, den, oldden

           Real(4)::  dumm, av, a, b,   maxkrigvar,&
                                       oldmaxkrigvar
!                Real::krigvarm
```

```
                Real(8):: nn001


           Integer::iv


                character(12), intent(in)::fokoutn



           rewind(unit=8)
                read(8,*)
                read(8,*) oldden, nn001,  maxkrigvar,  oldkrigvar


             oldmaxkrigvar=maxkrigvar


!              write(*,*)'maxkrigvarm: ', maxkrigvar
!                        write(*,*)'oldmaxkrigvarm: ', oldmaxkrigvar
!                        write(*,*)'sumdifn: ', sumdifn
!                        write(*,*)'maxsumdifn: ', maxsumdif
!                        write(*,*)'oldmaxsumdifn: ', oldmaxsumdif
!


! inicializacao das variáveis

            if(limit_counter.le.0) then
            oldden=variancia_inicial
                       den=variancia_inicial

            end if


                       call crosserror(krigvarm)

                       write(*,*) 'erro quadrático médio = ', krigvarm


                       if(krigvarm.gt.maxkrigvar) then

                          maxkrigvar = krigvarm


                       end if


! Definicao da funcao objectivo: se se pretende minimizar a sk2 e maximizar a redundancia,
! entao a funcao devera ser como abaixo

                       if(maxmin.eq.0) then

                       den=krigvarm/maxkrigvar*sk2_coef

! Se se pretender minimizar a redundancia e minimizar a redundancia, entao a f.o devera
! ser como abaixo

                       else

                       den=krigvarm/maxkrigvar*sk2_coef

                       end if


            object=den-Oldden*oldmaxkrigvar/maxkrigvar

                         if(object.lt.1e-6.and.object.gt.-1e-6) then

                           object=0.0
```

```
                        end if


           rewind(unit=8)
                  read(8,*)
                  write(8,*) oldden, nn001,  maxkrigvar,  &
                             oldkrigvar


!                     write(*,*) 'ln 924, oldden, summ,  maxkrigvar'
!                     write(*,*) oldden, summ, maxkrigvar



               close (6)

          END Subroutine Cost


! Subrotina onde se verifica se a fun‡Æo objectivo decresce, se sim o ponto
! , aceite; se nÆo o ponto , aceite de acordo com a expressÆo de Metropolis.
! DÆo entrada o custo estimado em "custo", a temperatura actual, tm, e a fun-
! ‡Æo randomize; Sai a vari vel l¢gica ansm.

         Subroutine metrop (obj,denm, olddenm,tm, ansm, sinalm)

           Implicit none

           Real(4), intent(in):: denm, tm, obj, olddenm
           Logical,Intent(out)::ansm, sinalm
           Real(4)::metro, h601, g502, cc803, fd504


                  open(unit=99, file='temp99.tmp')

            call gettim(hour, minute, second, p100second)


                     call seed(RND$TIMESEED)


!                    write(*,*) 'ln 726'


               call random(ra)

                     call seed(1000*int(ra*(p100second-60-second)**2/(ra*p100second)))
!

1022            call random(ra)

                              call random(ra)

                              call random(ra)

!            ra=ra*(p100second**2)/(45**2)
!                     ra=ra*(90**2)/(45**2)


!                    if(ra.lt.0.00001.or.ra.gt.1.0) then


!                              goto 1022
!
!                             end if


                     write(99,*)  ra

!                    write(*,*) 'seed:', 1000*int(ra*(p100second-60-
second)**2/(ra*p100second))
```

```
            if(limit_counter.le.0) then

                        summ=0


                      end if


                      metro=exp(-denm/tm)

!                     write(*,*) 'ln 694 N, nt, oldenm, tm: ', limit_counter, count_rep,&
!                                   olddenm, tm


            if(metro.gt.1.0) then

             metro=1.0

            end if

!           write(*,*) 'ln 723  denn, tm ', denm, tm, limit_counter, count_nlimit


                  ansm= obj <= 0.0 .or. ra <= metro

                   if(obj.lt.0.0) then

                        onlybettercounter=onlybettercounter+1

                      end if

!                     write(*,*) 'ansm= obj <= 0.0 .or. ra <= metro'
!                     write(*,*) 'ansm=',ansm,' obj= ',obj,' ra= ', ra,' metro= ', metro


               rewind(8)
                          read(8,*)
                          read(8,*) h601, summ, fd504,  hx06


                  if(ansm) then


                           count_rep=count_rep+1

                             if(count_rep.ge.rep) then


                                 sinalm=.TRUE.


                              end if

                              summ=summ+olddenm


                              oldkrigvar=krigvarm


                            rewind(8)
                            read(8,*)
                            write(8,*) denm, summ,  fd504,  krigvarm

                          else

                            rewind(8)
                            read(8,*)
                            write(8,*) olddenm,  summ, fd504,  hx06
```

```
                    end if

!                      write(*,*) h601, summ, fd504,  hx06
!                      write(*,*) denm, summ,  fd504,  krigvarm

                  write(8,*) ansm

                           close(99)

          END Subroutine metrop

! Subrotina onde se geram os vectores de pontos da rede a testar no algoritmo
! Para tanto , criado localmente um vector de dimensÆo igual … de xn e yn;
! Esse vector , alerado numa posi‡Æo a cada itera‡Æo, fazendo com que o os
! vectores xn e yn tamb,m sejam, e apenas na posi‡Æo definida por vector.

       Subroutine vectores (nv, ni, npontosv, xv, yv,k1v)

!             ,k2v,k3v,k4v)

         Implicit none

         Integer, intent(in)::nv, ni, npontosv
         Real(4), dimension(nv)::xnv, ynv
                Integer, dimension(nv)::k1nv
!               ,k2nv, k3nv,k4nv
         Real(4), dimension(npontos):: xv, yv
                Integer, dimension(npontos)::k1v
!               , k2v, k3v,k4v
         Real(4):: d1, d2, a, b, d301,  d303
         Integer, dimension(nv)::vector
         Integer::dummy1, dummy2,i, z, g,c, count
         Logical::ansv


                    rewind(8)


                    read(8,*)

!                   write(33,*) 'ln 755'

                    read(8,*) d1, a,  d301, d303


                    read (8,*) ansv



                    do i=1,ni+5

                      read(5,*)

                  end do



! Inicializa‡Æo aleat¢ria e carregamento do vector

                 if(limit_counter.eq.0) then


                                  rewind(9)
                                  rewind(10)
```

```
                                    s302: do i=1,nv


                                     call random(ra)

!                                           ra=ra*(p100second**2)/(45**2)

1023                                  if(int(ra*npontosv).lt.1) then

                                       call random(ra)

                                              goto 1023

                                     end if

1024                                  if(int(ra*npontosv).gt.npontosv) then

                                     call random(ra)

                                               goto 1024

                                end if


                           vector(i)=int(ra*npontosv)

                                              write(10,*) vector(i)
                                              write(9,*) vector(i)


                       end do s302


                                         goto 13

                                   end if



12              if(ansv) then              !Se a FO foi melhorada - guarda em
                                               !temp3 o novo vector (a partir de temp4)


                   rewind(9)
                   rewind(10)


                                  s404: do i=1,nv


                                         read(10,*) vector(i)                ! temp4.tmp
                                         write (9,*) vector(i)          ! temp3.tmp

                                   end do s404

                                  else



                    rewind(9)
```

```
                              s508: do i=1, nv
                                read(9,*) vector(i)


                  end do s508
                              goto 13
                         end if


                             call random(ra)

!                            ra=ra*(p100second**2)/(45**2)
1026                   if(int(ra*(nv)).lt.1) then
                               call random(ra)
                               goto 1026
                             end if
1027             if(int(ra*(nv)).gt.nv) then
                           call random(ra)
                               goto 1027
                         end if


!                  write(*,*) 'ln 980, ra', ra
                    dummy1=int(ra*(nv))


                    call random(ra)
!                               ra=ra*(p100second**2)/(45**2)

1028                   if(int(ra*(npontosv)).lt.1) then
                               call random(ra)
                               goto 1028
                              end if
1029            if(int(ra*(npontosv)).gt.npontosv) then
                             call random(ra)
                                goto 1029
                            end if


!                    write(*,*) 'ln 1006, ra', ra
```

```
                  dummy2=int(ra*(npontosv))


! Carregamento de um valor aleat¢rio na posi‡Æo dummy1, igual a dummy2,
! i.,, na posi‡Æo dummy1 do vector entra o valor dummy2. Filtrando a matriz
! de dados com vector obtem-se o novo vector a testar - identificado por xn e yn

13               s801: do z=1,nv

                 if(dummy1.eq.z) then
                   vector(dummy1)=dummy2

                 end if

                 end do s801



                   count=0

                s111: do g=1,nv-1
                  s112: do c=g+1,nv

                   if(vector(g).eq.vector(c)) then
                    count=count+1
                   end if

                  end do s112
                end do s111



                  if(count.ge.1) goto 12

! Controlo

                       do i=1,nv


                        end do

                                if(limit_counter.eq.0) then


                                     rewind (unit=9)
                                     rewind(10)

                                  s222: do z=1, nv

                                   write(9,*) vector(z)
                                       write(10,*) vector(z)


                                  end do s222

                                else

                                  rewind(10)

                                      s908: do i=1, nv

                                        write(10,*) vector(i)


                                   end do s908

                                end if
```

```
!                                   write(*,*) (vector(i),i=1,nv)



33                    xnv=xv(vector)
                     ynv=yv(vector)
                                  k1nv=k1v(vector)
!                                 k2nv=k2v(vector)
!                                 k3nv=k3v(vector)
!                                 k4nv=k4v(vector)



                                 s903: do z=1,nv


                                        write(5,*) xnv(z), ynv(z), k1nv(z)

!                                       , k2nv(z), k3nv(z), k4nv(z)



                                 end do s903




                                   rewind(8)

                                   read(8,*)
                                   write(8,*)d1, a,  d301,  d303
                                write (8,*) ansv

                                 close (5)
                                 close(9)


           END Subroutine vectores

                       Subroutine selsort(Bigvectors)

                                     Integer, dimension(:), intent(inout):: bigvectors
                                     Integer::numItems, smallestItem, I, Location_smallest
                                     Integer, dimension(1)::minloc_array

                                     NumItems=Size(bigvectors)

                                     do i=1, NumItems-1

                                            SmallestItem=MINVAL(Bigvectors(i:NumItems))
                                            Minloc_Array=MINLOC(Bigvectors(i:NumItems))
                                            Location_smallest=(i-1)+Minloc_array(1)

                                            Bigvectors(Location_smallest)=Bigvectors(i)
                                            Bigvectors(i)=smallestItem

                                     end do


                           end subroutine selsort


                           Subroutine crosserror(error2m)

                           Implicit none
```

```
                        Real::error2m
                        Real, dimension(n+ni):: ik1, ik2, ik3, ik4, true, error
                        Integer:: count_valid, internal_counter



                         ik1=0.0
                         ik2=0.0
                         ik3=0.0
                         ik4=0.0
                         true=0.0
                         error=0.0


                        error2m=0.0


                        read(18,*)
                        read(18,*)
                        read(18,*)
                        read(18,*)
                        read(18,*)
                        read(18,*)
                        read(18,*)




                                do i=1, n+ni

                                        read(18,*, err=1020) ik1(i), ik2(i), ik3(i), ik4(i),
true(i)

!                                       write(*,*) ik1(i), ik2(i), ik3(3), ik4(4), true(i)

                                        internal_counter=internal_counter + 1

                                        if(ik1(i).lt.-9998.or.ik2(i).lt.-9998.or.ik3(i).lt.-
9998.or.&
                                           ik4(i).lt.-9998) goto 1020


                                        count_valid = count_valid + 1

! Estimador por classes:

        if(ik1(i).ge.ik2(i).and.ik1(i).gt.ik3(i).and.ik1(i).gt.ik4(i)) then


                                        error(i) = 1.0 - true(i)

                                end if

                                if(ik2(i).ge.ik3(i).and.ik2(i).gt.ik4(i)) then

                                        error(i) = 2.0 - true(i)

                                end if

                                if(ik3(i).ge.ik4(i)) then

                                        error(i) = 3.0 - true(i)

                                        else

                                        error(i) = 4.0 - true(i)

                                end if
```

```
!Opção com uso do valor esperado

!                                    error(i) = ik1(i) + ik2(i)*2.0 + ik3(i)*3.0 +
ik4(i)*4.0 - true(i)

!                                    write(*,*) 'error(',i,')=', error(i)

                                     error2m= error2m + error(i)**2

1020                  write(*,*)

                             end do

                             error2m=error2m/(count_valid)

                      end subroutine crosserror


!                     subroutine uncertainty(N, M, K)

!                      integer:: n, m,
!                      real, allocatable, dimension(:,:):: x, v, r
!                      real, allocatable, dimension(:):: std
!                      character*1:: weight

!                      Allocate(x(n,m))
!                      Allocate(std(m))
!                      Allocate(xbar(m))
!                      Allocate(v(m,m))
!                      Allocate(r(m,m))

!                      weight='U'


9999          END Program crossvar
```